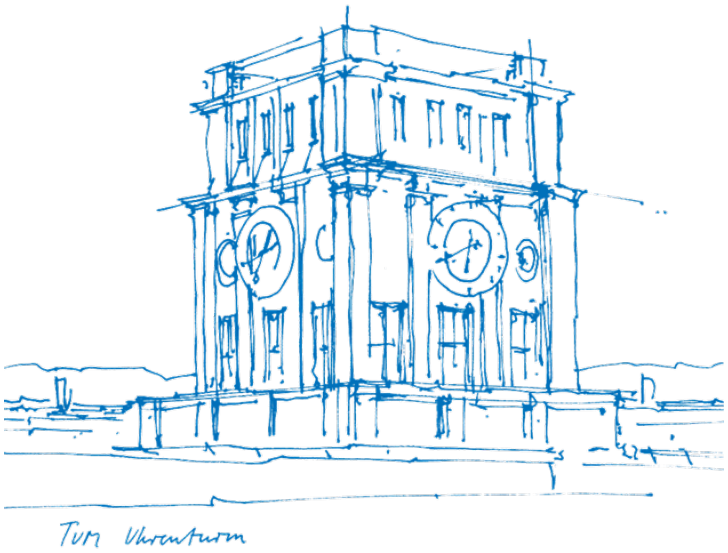
Artificial Intelligence in Healthcare and Medicine
TUM School of Computation, Information and Technology
Technical University of Munich

# Automated Point-of-Interest Prediction on CT Scans of Human Vertebrae Using Spine Segmentations

**Daniel-Jordi Regenbrecht**



# Automated Point-of-Interest Prediction on CT Scans of Human Vertebrae Using Spine Segmentations

Daniel-Jordi Regenbrecht

Artificial Intelligence in Healthcare and Medicine
TUM School of Computation, Information and Technology
Technical University of Munich

# Automated Point-of-Interest Prediction on CT Scans of Human Vertebrae Using Spine Segmentations

**Daniel-Jordi Regenbrecht**

Thesis for the attainment of the academic degree

**Master of Science (M.Sc.)**

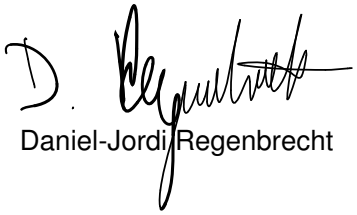
at the TUM School of Computation, Information and Technology of the Technical University of Munich.

**Examiner:**
Prof. Dr. Daniel Rückert

**Supervisor:**
Hendrik Möller

**Submitted:**
Munich, 15.05.2024

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.



Munich, 15.05.2024 Daniel-Jordi Regenbrecht

# Abstract

Back pain, a prevalent condition with complex and often poorly understood causes, presents significant challenges in medical diagnosis and treatment. Biomedical simulations offer a promising way to improve our understanding of this issue and facilitate patient-specific diagnosis and prognosis. However, the effectiveness of these simulations hinges on accurately determining key anatomical points, such as ligament attachment points on vertebrae. This thesis introduces a novel neural architecture to predict such anatomical Points of Interest (POIs). Utilizing spatial attention and transformer technologies, our model employs semantic segmentation masks instead of direct Computed Tomography (CT) inputs. Initial results demonstrate competent performance, suggesting that this approach could be viable for further refinement and clinical application.

Building upon this, we also explored the model's utility in surgical screw placement, a scenario challenged by a limited dataset. We attempted multi-task learning, training simultaneously on the ligament attachment point data and pedicle screw placement data; however, this approach did not yield performance improvements. In contrast, when addressing the larger, albeit noisier, and more inconsistent ligament attachment point dataset, we experimented with enhancing this dataset's consistency by employing a self-learning technique. While this method may introduce bias without expert oversight, it demonstrated that the model could learn more effectively from more uniformly annotated data. This finding suggests that improving annotation consistency might be vital in enhancing model performance.

This contribution adds to the body of knowledge advocating for the use of artificial intelligence to enhance clinical accuracy and efficacy and identifies areas for improvement.

# Contents

## List of Acronyms

**POI** Point of Interest

**CNN** Convolutional Neural Network

**SSC** Submanifold Sparse Convolution

**ALL** Anterior Longitudinal Ligament

**PLL** Posterior Longitudinal Ligament

**FL** Flaval Ligament / Ligamentum Flavum

**SSL** Supraspinous Ligament

**ISL** Intraspinous Ligament

**ITL** Intertransverse Ligament

**CT** Computed Tomography

**ViT** Vision Transformer

**UNETR** U-Net Transformer

**GCN** Graph Convolutional Network

**MLP** Multi-Layer Perceptron

**LSTM** Long Short-Term Memory

**ReLU** Rectified Linear Unit

**LiAP Dataset** Ligament Attachment Point Dataset

**PS Dataset** Pedicle Screw Dataset

**MSE** Mean Squared Error

**CDF** Cumulative Distribution Function

**NN** Neural Network

# 1 Introduction

Back pain remains one of the most prevalent and debilitating medical conditions, affecting millions worldwide. As reported by the 2010 Global Burden of Disease Study, low back pain is the single leading cause of years lived with disability globally, impacting 9.4% of the population [38]. The underlying causes include both traumatic [52] and degenerative [76] spine conditions, which can often be effectively treated with surgery [21]. However, despite its widespread impact, the causes of low back pain remain poorly understood in many cases, leading to ineffective treatment strategies [35].

## Motivation

Recent advancements in biomedical simulations have significantly contributed to our understanding of spinal health. These simulations utilize finite element analysis to model the mechanical behaviors and stress distributions along the spine under various physiological conditions [55, 80]. However, the effectiveness of these simulations is heavily reliant on the precision with which individual patient data, particularly ligament attachment points, are modeled and analyzed. Although current imaging techniques, such as CT, offer detailed insights, the manual identification of specific vertebral points remains time-consuming and prone to errors [15]. This highlights the necessity for automated methods to enhance the accuracy and efficiency of spinal analysis, ensuring that simulations are comprehensive and personalized.

A critical technique in spinal surgery is the placement of pedicle screws, employed to achieve vertebral stabilization and correct deformities. The precision of screw placement is crucial, as inaccuracies can lead to severe complications, including neurological damage. Increasingly, guided CT techniques and integrating robotic systems in surgical procedures are being explored to enhance the precision and safety of these operations [59]. This growing research interest in robotic surgery underscores the importance of developing robust methods for accurate anatomical landmark identification, which is essential for guiding these advanced technologies.

Automated point-of-interest (POI) prediction in medical imaging is rapidly evolving, embedded within a broader trend of enhancing the speed and depth of data interpretation in healthcare through artificial intelligence [43, 95]. This trend, coupled with significant advancements in deep learning-enabled computer vision, increasingly improves patient care and clinical workflows [18]. While these advancements promise substantial improvements in diagnostic and operational efficiency, their application in precision-critical tasks such as spinal health modeling and surgical planning warrants detailed exploration.

## Objectives, Scope, and Outline of the Thesis

The primary objective of this thesis is to develop a novel neural architecture that enhances the prediction of anatomical points of interest, focusing mainly on ligament attachment points for spine simulations. This involves designing a model that autonomously identifies and annotates vertebral landmarks from CT images using advanced semantic segmentation techniques. Additionally, this research explores the feasibility of applying the developed model to automatic pedicle screw placement in spinal surgery. Through this work, we aim to investigate whether such methods can potentially influence future biomedical simulation and orthopedic surgery practices, contributing to the ongoing dialogue on improving patient-specific treatment outcomes.

The second chapter of this thesis is dedicated to introducing the medical background and theoretical foundations of the techniques used in this work. Following this, the third chapter reviews existing methods

in anatomical landmark prediction. Chapter 4 introduces the datasets used in this work, proposes a novel architecture for anatomical landmark prediction, and explains the loss functions and methods we choose to evaluate our work. In the fifth chapter, we motivate and describe several experiments to gain insight into the viability of our process for our objectives. We will discuss our findings and encountered limitations in chapter 6 and conclude this work in chapter 7 with a summary of this work and its results, focussing on future directions and impact.

# 2 Background

This chapter outlines the thesis's key concepts and theoretical foundations. We begin with the medical concepts, followed by foundations in deep learning for computer vision, which we build upon. We conclude the chapter by reviewing two classical image-processing tasks: segmentation and registration.

## 2.1 Medical Background

In this section, we focus on the medical context of this work: The anatomical structure of the spine and individual vertebrae are explained, followed by details on pedicle screw placement.

### 2.1.1 The Vertebral Column and Spine Regions

The human spine, a complex anatomical structure, serves multiple critical functions, including protecting the spinal cord, supporting the body's weight, and enabling flexible movement. The individual bones that make up the spine, also called the vertebral column, are called vertebrae. The spinal column is divided into five distinct regions: The cervical, thoracic, and lumbar spine, the sacrum, and the coccyx. (see Figure 2.1) While the cervical, thoracic, and lumbar spine vertebrae are distinct and connected only by soft tissue, the os sacrum and coccyx consist of several fused vertebrae. Other than the highly specialized first and second cervical vertebrae, the other vertebrae share the following common substructures (see Figure 2.2):

- **Spinous process:** A bony projection off each vertebra's posterior (back) part that can be felt through the skin. It serves for muscle attachment and ligament attachment.
- **Transverse process:** A small bony projection on either side of the vertebral arch where the lamina meets the pedicle. It serves as a point of attachment for muscles and ligaments.
- **Vertebral arch:** The arch formed from joining all posterior extensions from the vertebral body (pedicles and laminae), enclosing the vertebral foramen.
- **Pedicle:** A short, stout process that extends posteriorly from the dorsal part of the vertebral body, connecting the vertebral body to the transverse processes and laminae.
- **Lamina:** The part of the vertebral arch that forms the roof of the arch and helps complete the enclosure of the vertebral foramen.
- **Vertebral body:** The thick, cylindrical part of a vertebra that forms the anterior portion of the vertebral structure and provides strength and support to the spine.
- **Articular process:** A set of projections that arise from the vertebral arch, comprising superior and inferior articular processes, which help form the facet joints between the vertebrae, allowing for spine mobility.

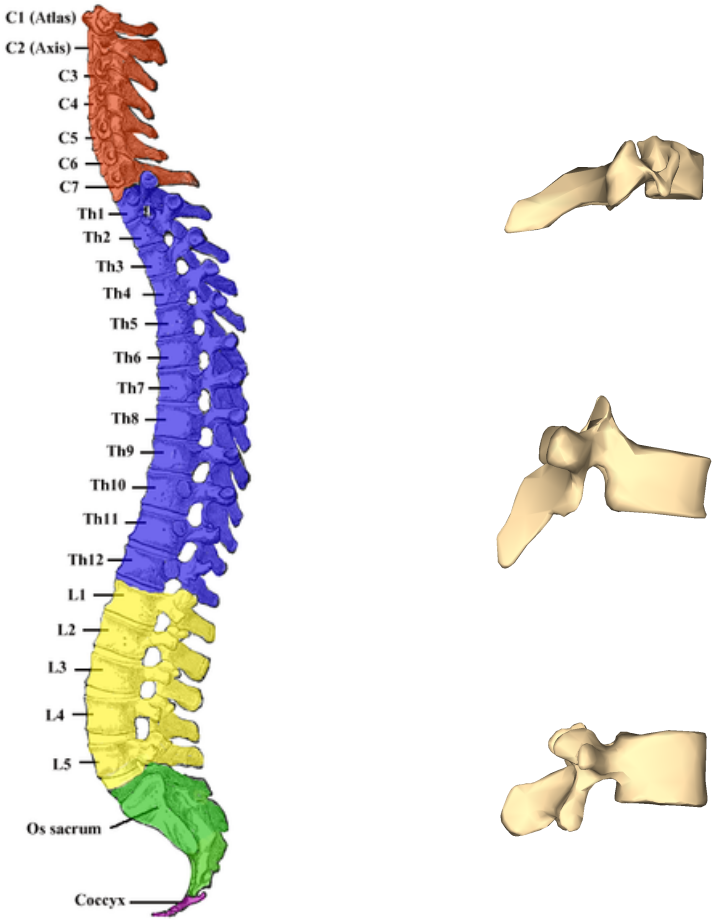



**Figure 2.1** Left: Overview of the vertebral column and its segments (taken from [84]. Right, from top to bottom: Typical cervical vertebra, thoracic vertebra, and lumbar vertebra (taken from [64].

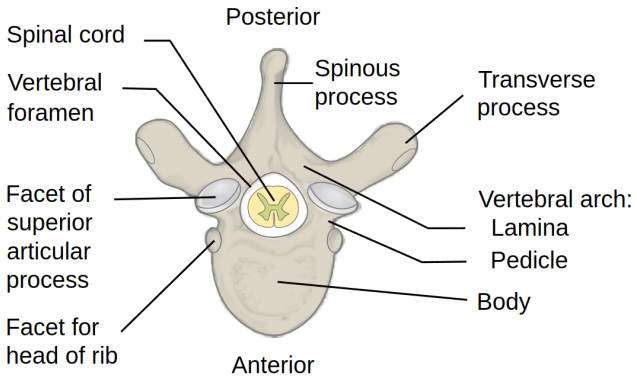



**Figure 2.2** Substructures of a vertebra, viewed from above (taken from [14]).

Generally, vertebrae within one of the subregions share some common characteristics between them, distinguishing them from vertebrae in other spine regions:

- **Cervical Vertebrae**: Comprising seven vertebrae (C1-C7), the cervical spine is highly flexible, supporting the head's rotational movements. The atlas (C1) lacks a vertebral body and forms a ring that supports the skull, while the axis (C2) features a peg-like projection called the odontoid process that pivots within C1. The other cervical vertebrae possess small, wider bodies, larger triangular vertebral foramina (openings), bifid (split) spinous processes, and transverse foramina for the passage of the vertebral artery.

- **Thoracic Vertebrae**: Including twelve vertebrae (T1-T12), the thoracic spine serves as the attachment point for ribs, encasing vital organs. Thoracic vertebrae have larger bodies than cervical but smaller than lumbar vertebrae, with longer and downward-sloping spinous processes to facilitate limited flexion and extension. Unique features include smooth, flat surfaces on the vertebrae where the ribs attach called costal facets on the transverse and body for rib articulation, contributing to restricted rotational ability.

- **Lumbar Vertebrae**: Consisting of five vertebrae (L1-L5), the lumbar spine bears the body's weight and supports forceful movements. Lumbar vertebrae are the largest, designed to sustain significant loads with broad, short spinous processes and massive, kidney-shaped bodies. They lack transverse foramina, and their articular facets are oriented to allow flexion and extension while limiting rotational movements.

While these general shared characteristics make the distinction of the spine regions relevant and useful, there is still significant variation in the shapes of the vertebrae within the same region, depending on their position in the spinal column. As an example, the T1 vertebra has a medium-sized body with a more cervical-style, downward-sloping spinous process, while the T12 vertebra has a larger body typical of thoracic vertebrae but features a longer, more horizontal spinous process transitional to the lumbar style. Further, the morphology of the same vertebra level can differ between individuals due to variations in bone density, vertebral size, and shape influenced by genetic factors, aging, and individual health conditions.

### 2.1.2 Ligaments of the Spine

Ligaments are essential stabilizers of the spine, contributing to its structural integrity and functional flexibility. Six spine ligaments are considered in this thesis. The following overview is based on the descriptions of Panjabi *et al.* [70] and Johnson [44] *et al.*.

- **Anterior Longitudinal Ligament (ALL)** The ALL extends along the vertebral bodies' anterior surface from the skull's base to the sacrum. It is integral in limiting hyperextension of the spine and providing stability. The ligament is wider and thinner in the cervical region and becomes thicker and narrower as it descends towards the lumbar spine. It is crucial for preventing vertebral displacement and maintaining spinal alignment.

- **Posterior Longitudinal Ligament (PLL)** Positioned within the vertebral canal along the posterior side of the vertebral bodies, the PLL is narrower and thicker than the ALL. It runs from the second cervical vertebra down to the sacrum. This ligament plays a key role in limiting the spine's flexion and contributes to stabilizing the intervertebral discs.

- **Flaval Ligament** / **Ligamentum Flavum (FL)** These ligaments connect the laminae of adjacent vertebrae from the axis (C2) to the sacrum. They are rich in elastic fibers, which help preserve the spine's natural curvature and assist in straightening the spine after flexion. The ligamentum flavum is a barrier that protects the spinal canal's neural elements and helps prevent excessive flexion.

- **Intraspinous Ligament (ISL)** Situated between the spinous processes of adjacent vertebrae, the interspinous ligament extends from the cervical to the lumbar spine. It assists with limiting spinal flexion and provides proprioceptive feedback, which is essential for spinal stability and coordination.
- **Intertransverse Ligament (ITL)** Found between the transverse processes of the vertebrae, these ligaments vary in thickness and length across different spinal regions. In the thoracic region, they are fibrous and well-developed, while in the lumbar region, they are thin and membranous. They play a role in limiting lateral flexion of the spine.
- **Supraspinous Ligament (SSL)** This ligament connects the tips of the spinous processes from the seventh cervical vertebra down to the sacrum. It helps limit the spine's flexion, maintain alignment, and distribute loads along the spine. The supraspinous ligament provides a continuous attachment site for muscles and fascia across the posterior spine, contributing further to spinal stability.

The ALL, PLL, and FL all have a significant width, which varies not only between different types of vertebrae along the spine but also from patient to patient [33]. The widths of these ligaments also bear important implications for the biomechanical properties of the spine, e.g., their stiffness is determined in large parts by their width, with a wider ligament generally offering more resistance to deformation [94, 49]. Further, for each vertebra, the corresponding ligament segments cover a significant area on the surface instead of attaching at a single point. The other ligaments are much finer structures, and the contact of area can be reasonably described by a single point.

### 2.1.3 Pedicle Screws

Pedicle screws are specialized screws used in spinal surgery to create a rigid connection between spinal segments, helping to stabilize the spine and facilitate fusion. These screws are inserted into the pedicle of the vertebrae, the dense, cylindrical structures that connect the vertebral body to the vertebral arch. Pedicle screws are utilized in various conditions, including spinal deformities like scoliosis and spondylolisthesis, traumatic fractures, and spinal instabilities or degeneration.

The insertion of pedicle screws is a technically demanding procedure that requires precise placement to avoid damage to the spinal cord or nerve roots. Traditionally, surgeons use either a freehand technique, fluoroscopy-assisted, or computer navigation methods to improve placement accuracy. Computer navigation has shown higher accuracy rates, particularly in complex areas such as the thoracic spine, which has a higher risk of screw misplacement due to the smaller size of pedicles. Techniques like computed tomography (CT) navigation or robot-assisted placement are often preferred for their precision in these challenging areas [73, 63].

Studies have highlighted the critical importance of proper pedicle screw placement to prevent postoperative complications. Techniques and tools have continually evolved to enhance the safety and efficacy of this procedure, reducing the risks associated with screw misplacement and improving overall surgical outcomes [50].

Evaluating the quality of pedicle screw placement, however, is a widely discussed problem with no definitive answer [50, 41, 74, 63]. One of the universally accepted key properties of accurate screw trajectories is that they are firmly placed within the pedicle with no breaches of the vertebra surface as these entail the risk of damaging surrounding structures [50, 41]. Figure 2.3 shows an ideal screw trajectory (Category Ia in the Zdichavsky Scale [96]) compared to cases of lateral and medial breaching.

## 2.2 Deep Learning Methods in Computer Vision

Convolutional Neural Networks (CNNs) is currently the primary method for analyzing spatiotemporal data such as images and 3D volumes, including in the realm of medical image data [2, 58]. In this section, we will introduce the convolutional operator that forms the basis of all CNNs, as well as the pooling operator

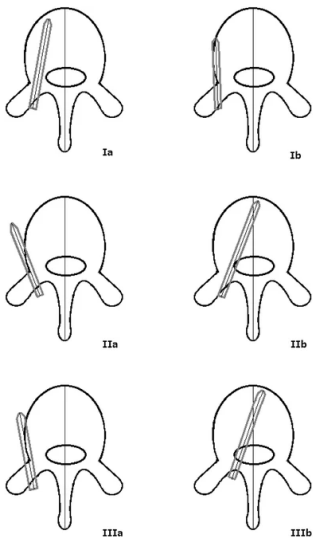



**Figure 2.3** Zdichavsky Grades: Overview of well-placed (Grade Ia) and various categories of misplaced screws, penetrating either the lateral surfaces or the spinal canal. Taken from [41]

that almost all CNNs employ [26]. Finally, we review recent architectures focusing on DenseNet [39], a variant of which we will explore in this work.

### 2.2.1 The Convolutional and Pooling Operators

The convolutional and pooling operators have been used in image processing since long before the advent of deep learning in signal processing [8]. In traditional applications, though, the convolution parameters were handcrafted, requiring significant labor and limiting adaptability. A significant paradigm shift came with LeCun *et al.*'s application of backpropagation to convolutions in 1989 [53], which introduced learnable parameters in convolutional layers, enhancing adaptability during training. This innovation led to the CNN development with multiple convolutional and pooling layers [54]. Following Goodfellow *et al.* [26] this chapter lays out the mathematical definition of the convolutional operator and how it has been adapted to convolutional layers that are the cornerstone of CNNs.

**The Mathematical Definition of Convolution** The convolutional operator in CNNs is inspired by and named after the convolutional operator stemming from functional analysis, with applications ranging from signal processing to probability theory. Mathematically, the convolution of a function $f$ with another function $g$ is defined as

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.1}$$

It is common in this context to refer to the function $f$ as kernel and the function $g$ as input. By interpreting an image, 3D volume, or similar grid-like data structure as a discrete function $g$, mapping an index of an element of the grid such as a pixel to an intensity value, we can apply a discretized, multidimensional variant of the convolutional operator. We will define this operator for the 2D case for simplicity, which corresponds to grayscale images. The extension to higher dimensions, as encountered in 3D data or multi-channel input, is straightforward. For a discrete 2D input $g$ and a discrete 2D kernel $f$, the convolutional operator is defined as

$$(f * g)(i,j) := \sum_{m\in\mathbb{Z},n\in\mathbb{Z}} f(m,n)g(i-m,j-n) \tag{2.2}$$

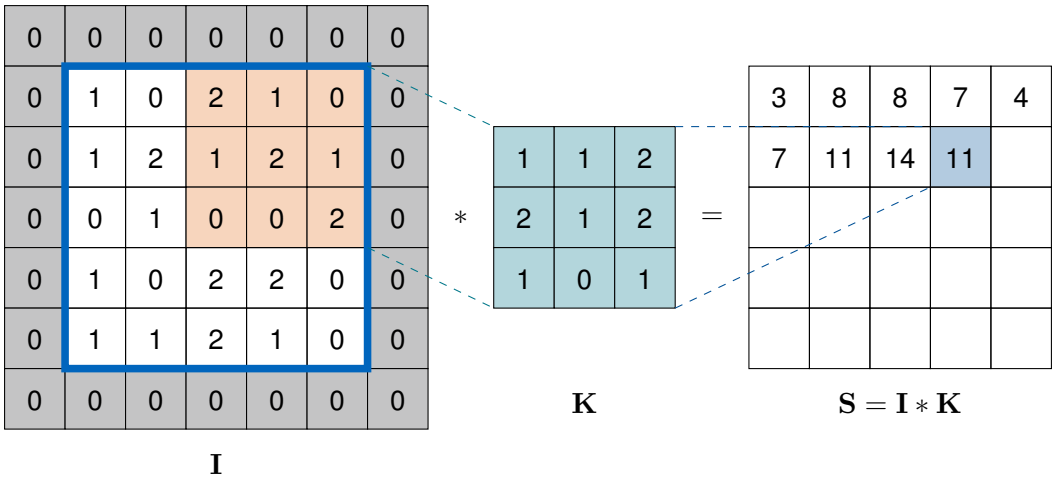



**Figure 2.4** Convolution of an Image with zero-padding I with a kernel K (adapted from [77]).

While the discretization and extension to more than one dimension do bring us one step closer to the convolutional operator as it is implemented in modern neural network frameworks, it is still not applicable in this form: Since the summation is unbounded it is impossible to evaluate on a computer and the image $g$ is undefined for indices outside of the image bounds, like negative values. In practice, we restrict the range of $m$ and $n$ to rather small values, typically much smaller than the image. E.g., it is common to use a kernel size of three, meaning that $m$ and $n$ each only take on the three distinct values $\{-1, 0, 1\}$. In this case, it suffices to represent the kernel $f$ as a three-by-three grid, containing the values for all possible combinations of $m$ and $n$ Further, the subtraction of the arguments in the image $g$ can be replaced with a summation without loss of generality, since this formulation is equivalent to transposing the kernel $f$, which does not matter since it is learned by the network anyway. Summation has been preferred because it is more straight-forward and leads to a nice interpretation of high outputs of the convolutional operator: They correspond to centers of image patches, where the kernel matches the image well, i.e., high values in the kernel overlap with high-intensity values in the image and vice versa. This makes the "convolution" operator a cross-correlation in a mathematical sense. Inserting the finite summation boundaries and summation instead of subtraction into Eq. 2.2, we arrive at the formulation of a three-by-three convolution corresponding to what is actually implemented in modern CNNs. We will denote the output of the operation as $S$, the input image as $I$, and the kernel as $K$ for clarity, the definition, however, still works for general discrete functions:

$$S(i,j) := \sum_{m,n \in \{-1,0,1\}} K(m,n) I(i+m, j+n) \tag{2.3}$$

A common technique to handle undefined values, e.g., for $m = n = 0$, $i = j = -1$, where we would use the value $I(-1,-1)$, which is not defined, is to set them to zero. This method is called zero-padding.

**Convolution as a Sliding Kernel** From this definition, we can understand how convolution can be viewed as "sliding" the kernel over the input image, measuring the cross-correlation at each point. Consider Figure 2.4: On the left, we see a five-by-five grid inside the blue box, which is the input and is padded with zeroes as explained above, and a three-by-three kernel is given. Note that the image indices are of the format (row, column), starting with zero, and the indices in the kernel are relative to the center of the kernel, i.e., the upper left entry in the kernel is $K(-1,-1)$. This aligns with our notation so far and with the convention that a convolutional kernel is always applied centered at its current location. In the figure, the kernel is at position $(1,3)$ of the image, to get the output $S(1,3)$, we apply Eq. 2.3: Each value in the orange patches, which are those that are processed due to the size of the kernel, is weighted by the corresponding value in the kernel $K$ and the results are summed up together. By "sliding" the kernel over the image, i.e., applying it in a way that the center is over each pixel in the input once, we get the overall output $S$.

**Pooling as a Spatial Aggregation** Following the convolution process, pooling, also known as subsampling or down-sampling, simplifies the output by reducing its spatial size, thereby preserving only the most essential features. In a typical setup, consider a convolutional layer outputting a four-by-four grid. Pooling operates over this grid with a common two-by-two window, moving this window across the grid with a stride of two. Each cell in this window undergoes a specific pooling operation—most commonly, maximum pooling, where the highest value within the window is selected, or average pooling, averaging the values within the window.

For instance, if the top-left window consists of the values $1, 2, 3, 4$, the maximum pooling operation selects $4$. This process effectively reduces the dimensionality of the data, decreasing computational requirements for subsequent operations and enhancing the network's robustness to variations in the position of features within the input. The output from this operation is a smaller grid, such as a two-by-two matrix, representing the maximum values from each non-overlapping two-by-two window across the original grid. This operation does not involve weight parameters but relies purely on the spatial structure of the data, emphasizing the most pronounced features and preparing the network for further processing layers.

### 2.2.2 DenseNet

DenseNet, introduced by Huang *et al.* [39] has achieved state-of-the art performance in several recent medical computer vision tasks [20, 87, 100]. As CNNs evolved, increasingly deep architectures were explored, from the 5 layers of the foundational LeNet proposed by LeCun *et al.* [54], the same number of convolutional layers Krizhevsky *et al.* [51] employ in their AlexNet, to the VGG using 16 to 19 layers [83] and the ResNet proposed by He *et al.* featuring several hundreds of layers [37]. When stacking convolutional layers to such depths without further measures, however, CNNs expose a problem dubbed "Vanishing Gradients": During backpropagation, due to the multiplicative nature of the gradient, the depth makes it likely that gradients will approach zero in the first layers, being the last in the gradient flow, effectively halting the network from learning further [37]. Depths like those in ResNet could thus only be achieved through the use of residual connections, providing shortcuts to the feature flow: Instead of taking only the output of a layer as the next-deepest feature, this output is added to the input. The gradient, therefore, flows along the summation in addition to the layer, ensuring a greater learning signal. DenseNet builds upon this idea by providing shortcuts of shallower feature representations to not only one but multiple deeper layers through concatenation within one so-called "Dense Block". Multiple of these Dense Blocks, with intermediate convolution and pooling layers, are stacked and followed by a pooling and a final linear layer to perform image classification in the original formulation. Figure 2.5 shows an overview of the entire architecture.

Five parameters define the exact architecture of a DenseNet:

- **Growth Rate**: Determines the number of feature maps added with each layer in a Dense Block. This parameter directly affects the increase in complexity and the total number of parameters in the network as layers progress.
- **Block Configuration**: An array of integers where each integer specifies the number of layers in a corresponding Dense Block. The overall depth and structure of the DenseNet are influenced by this configuration.
- **Number of Initial Features**: Specifies the number of output channels for the first convolutional layer before entering the first Dense Block, setting the initial complexity and the dimensionality of feature maps.
- **Bottleneck Size**: A multiplier for the growth rate that defines the number of channels in the bottleneck layers. These layers use 1x1 convolutions to compress the input feature maps, reducing computational load before more expansive operations.
- **Compression Factor**: Used in transition layers between Dense Blocks to reduce the number of feature maps by a factor (e.g., 0.5), which helps control the model's size and computational efficiency.

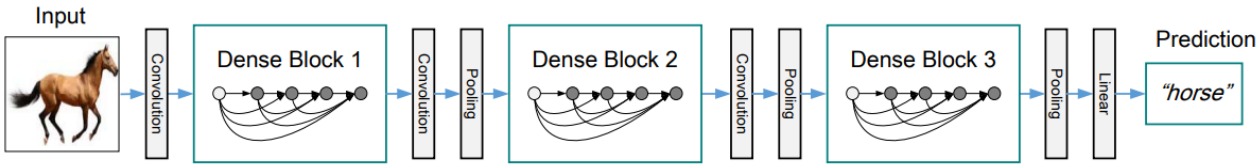



**Figure 2.5** The DenseNet architecture (taken from [39]. The lower arrows within the DenseBlocks represent the skip connections of features. The dots within the DenseBlocks represent a composition of batch normalization [42], Rectified Linear Unit (ReLU) [25] and a $3 \times 3$ convolution.

### 2.2.3 Attention Mechanisms and Transformers

The concept of attention, inspired by the tendency of humans to selectively concentrate on selected parts of information instead of processing all at once, has become one of the most intensely studied and quickly advancing aspects of deep learning of the past years [68, 9]. The core idea of attention is to allocate resources efficiently while reducing the complexity of the task, by highlighting important features while suppressing irrelevant ones. This paradigm can be applied in different contexts and on different levels [68]:

- **Spatial Attention** refers to selectively focussing on specific spatial locations within data such as images or volumes, dynamically prioritizing locations where important features are expected to occur.
- **Self-Attention** is a mechanism that enables a model to weigh the importance of different elements within a collection, e.g., a sequence or set, when processing any specific element. This process highlights how each element interacts with others and itself, hence the name "self"-attention.
- **Channel attention** enables models to emphasize certain channels (like color channels in images or deep feature channels) over others. It assesses the relevance of each channel to the task at hand, allowing the model to adjust its focus on more informative features.

This work will focus on the first two forms of attention mechanisms, Spatial Attention and Self-Attention. The latter forms the basis of the transformer architecture we will explore. The following two sections provide the theoretical foundations of these mechanisms:

**Spatial Attention** Spatial attention can be generally divided into two categories [68]: Hard attention refers to an "either-or" approach where, for a given image or feature map, certain regions or pixels are maintained, while others are set to zero, essentially modeling the question: Is this location important for the task? On the other hand, soft attention uses a pixel-wise score that assigns a weight to each location, aligning with the question: How important is this location for the task? Recent works incorporate soft spatial attention into CNNs and show improved performance with its use [90, 92]. These methods compute a heatmap, an attention mask, dynamically weighing the pixel-wise importance, multiplying element-wise ($\otimes$) to refine feature maps. Woo *et al.* [92] accordingly define (2D) spatial attention as follows: For a given feature map $F \in \mathbb{R}^{C \times H \times W}$, where $C$ denotes the channel depth and $H, W$ are the spatial dimensions and a spatial attention map $M_C \in \mathbb{R}^{1 \times H \times W}$, the resulting feature map is computed as

$$F' = M_S(F') \otimes F \tag{2.4}$$

where $M_S$ is copied along the channel dimension to match the shape of $F$. Figure 2.6 visually depicts how this operation affects a feature map.

**Self-Attention and Transformer** Bahdanau *et al.* [7] introduced self-attention as a mechanism for neural machine translation. The motivation came from the observation that some parts of a sentence are more relevant than others for translating individual words. Therefore, it is beneficial to equip a translation model

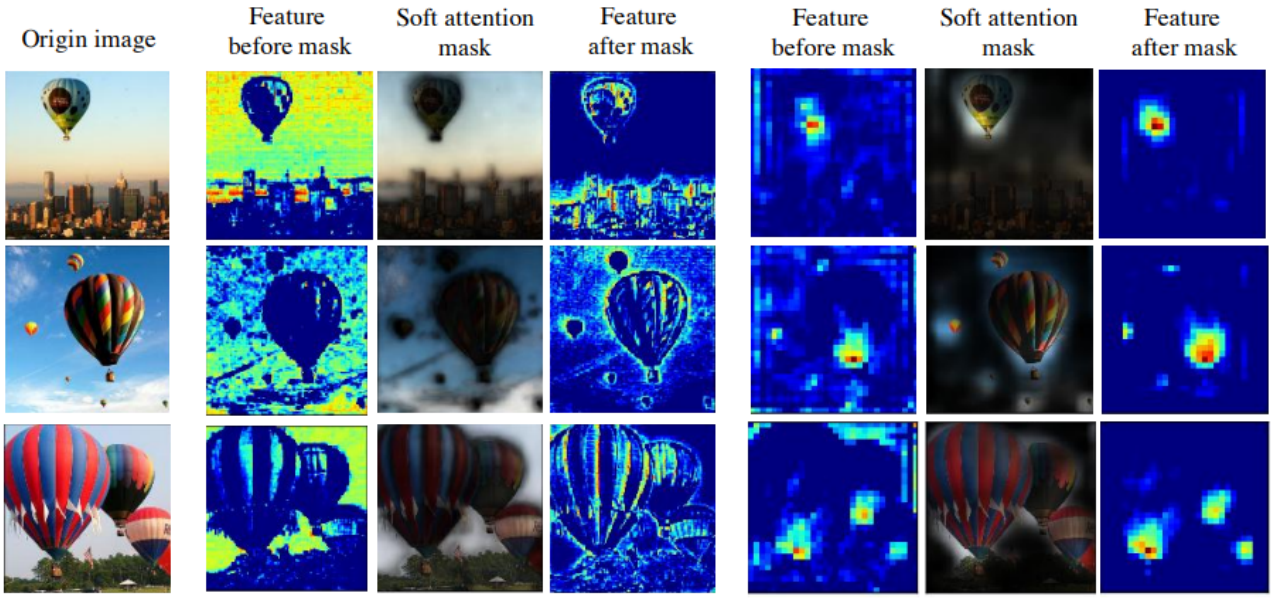



**Figure 2.6** Attention masks highlighting different locations, generating more pronounced feature responses (taken from [90]).

with a method to search for relevant words and aggregate their context automatically. For this, the self-attention mechanism was developed, which we will describe here, following the notation of Kamath *et al.* [45].

Overall, a self-attention layer takes in a list $x_1, ..., x_l \in \mathbb{R}^d$ of feature vectors and converts it to the output vectors $z_1, ..., z_l \in \mathbb{R}^{d_v}$. For each $i \in 1, ..., l$, a so-called query $q_i \in \mathbb{R}^{d_k}$, key $k_i \in \mathbb{R}^{d_k}$ and value $v_i \in \mathbb{R}^{d_v}$ is computed with the weight-matrices $W_q \in \mathbb{R}^{d \times d_k}$, $W_k \in \mathbb{R}^{d \times d_k}$ and $W_v \in \mathbb{R}^{d \times d_v}$, respectively:

$$q_i = W_q x_i \tag{2.5}$$
$$k_i = W_q x_i \tag{2.6}$$
$$v_i = W_v x_i \tag{2.7}$$

These weight matrices are initialized randomly and jointly learned during training along with other network parameters. The queries and keys are then used to compute the attention weights, i.e., the predicted relevance of the information associated with a token at position $j \in 1, ...l$ for the currently processed token at position $i$. This is done in two steps: First, the unnormalized attention weight $\omega_{i,j}$ is computed as the dot-product, which is why the queries and keys necessarily have the same dimensionality (resulting from the same shape of the weight matrices by definition):

$$\omega_{i,j} = \frac{q_i^T k_j}{\sqrt{d_k}} \tag{2.8}$$

The motivation for the dimension-normalization term $\frac{1}{\sqrt{d_k}}$ is to eliminate the effect of the key/query-embedding dimensions: Larger dimensions lead to larger dot products. This can hurt training stability, as the next step involves taking the soft-max which produces more extreme values for larger inputs as pointed out by Vaswani *et al.* [88]. The calculation of the soft-max along the key dimension $j$ is defined as:

$$\alpha_{i,j} = \frac{e^{\omega_{i,j}}}{\sum_j e^{\omega_{i,j}}} \tag{2.9}$$

This ensures that the resulting attention weights are strictly positive and sum up to 1, essentially providing a distribution of the relative impact of each token at position $j$ for the processing of the current token

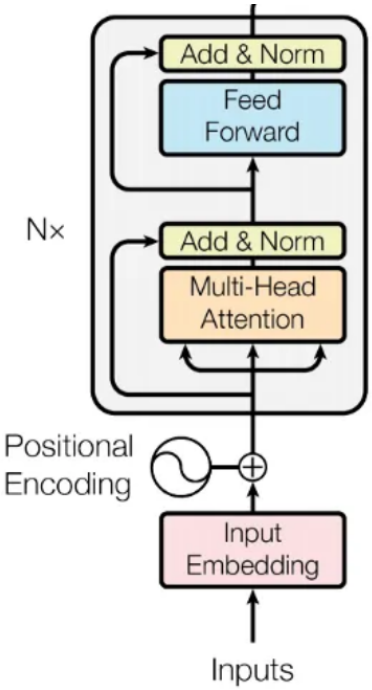



**Figure 2.7** The Transformer Encoder (taken from [88]).

at $i$. Notice that, in particular, each token computes an attention value to itself, which is not treated separately from the others, i.e., the attention needs to be shared across the token itself and its peers. This is where the term self-attention comes from. Since all operations in the self-attention block can be written as matrix-vector operations, the process can be executed in parallel for all position pairs, making self-attention particularly efficient. Usually, multiple instances of the above-described mechanism are executed in the same layer, and the outputs are concatenated; this is referred to as multi-head attention [88].

The self-attention mechanism lies at the heart of the transformer, presented by Vaswani *et al.* [88], also in the context of natural language processing. Their transformer encoder (see Figure 2.7 consists of several layers, each consisting of multi-head self-attention wrapped by a skip connection, subsequent layer normalization, and a simple feed-forward block, also with a skip connection and layer norm, i.e., a shifting and rescaling along the feature dimension to ensure a mean of $0$ and standard deviation of $1$ afterward. This normalization technique was proposed by Ba *et al.* and shown to improve training stability [6].

Introduced by Dosovitskiy *et al.* [17], the Vision Transformer (ViT) presents an adaptation of the transformer to computer vision tasks: The ViT, originally developed for image classification, divides the input image into separate patches, which are then fed into a transformer encoder along with a class embedding. The transformed class embedding feature vector is used in an Multi-Layer Perceptron (MLP) head to make the final predictions, while the transformed patch features are ignored. A different transformer-based architecture used in computer vision, specifically in the field of medical images, is the U-Net Transformer (UNETR) by Hatamizadeh *et al.* [36], which uses the transformer-encoded image patches for the down-stream task of segmentation (see Section 2.3). Both architectures successfully used transformers for Computer Vision, forming part of a larger trend in using transformer-based architectures in this domain [47].

## 2.3 Image Segmentation

Segmentation in digital imaging is a fundamental process that involves distinguishing and separating distinct areas within an image. This procedure assigns unique labels to each pixel or voxel, thereby creating what is known as a segmentation mask. The technique can be categorized into two main types (see Figure 2.9): instance segmentation and semantic segmentation. Instance segmentation identifies each specific

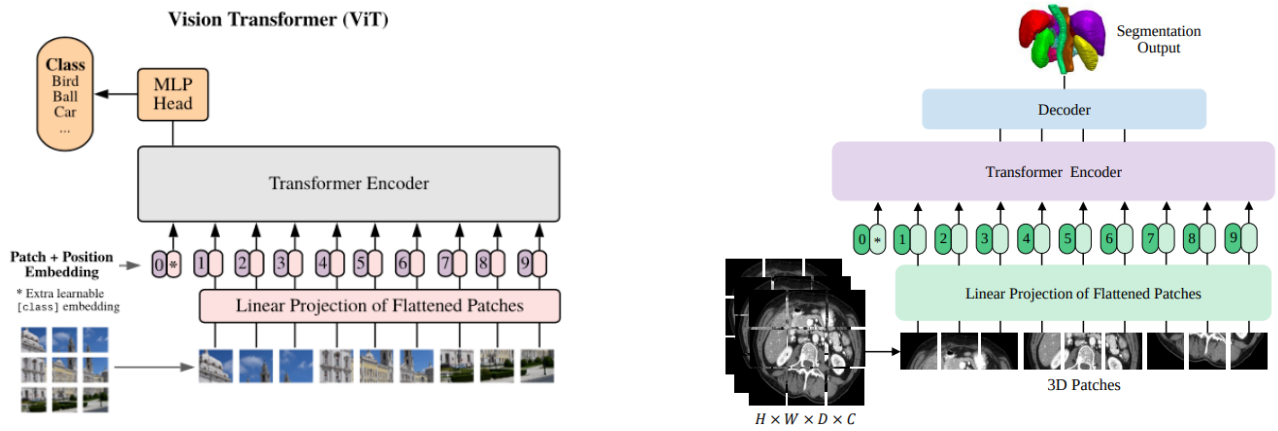



**Figure 2.8** The Transformer Encoder in ViT (left, taken from [17]) and UNETR (right, taken from [36])

object instance within an image, enabling the separate analysis of each entity. In contrast, semantic segmentation groups pixels into classes based on the type of object they represent, without differentiating between individual instances of the same class. These segmentation methods are critical for various applications, including automated analysis tasks such as object localization, volume measurement, and extracting geometric and morphological properties.

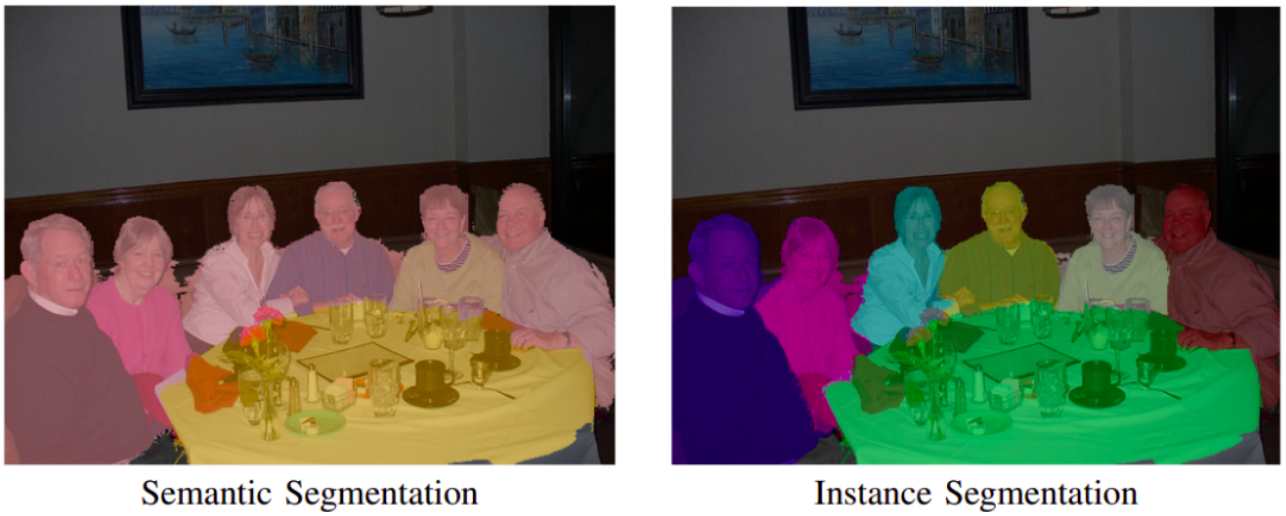



**Figure 2.9** Semantic Segmentation vs Instance Segmentation (taken from [5]).

Segmentation can be categorized into several types based on the methodology:

- **Manual Segmentation:** Involves the operator manually tracing the contours on each slice. It is accurate but time-consuming.
- **Semi-automatic Segmentation:** Combines manual initiation with software-propagated segmentation, reducing variability and workload.
- **Automatic Segmentation:** Utilizes fully automated algorithms, including neural networks trained on large datasets.

In the past decade, significant advances have been made in segmentation using CNNs. A cornerstone architecture developed specifically for semantic segmentation in biomedical images is the U-Net [78], featuring a design resembling the letter "U," which includes a contracting path to capture context and a symmetric expanding path that enables precise localization. This design helps the network to work

effectively with fewer training images and to excel at tasks where the prediction requires localization and context from the image, such as segmenting tumors from medical scans. Inspired by the success of the U-Net, several architectures have expanded on it by including recurrent residual convolutional layers [1], multi-resolution input [40], dense connections [98] or spatial attention [30].

## 2.4 Image Registration

Image registration is another fundamental task in computer vision, consisting of aligning two images in a common coordinate system such that corresponding points are overlaid at the same coordinates. To achieve this, one image, the reference image, is kept unchanged. In contrast, the second image, referred to as the moving image, is transformed to achieve the best fit to the reference in terms of a suitable similarity metric [28]. The transformations used can be categorized into [31]:

- **Rigid Transformation:** Preserves distances and angles, involving only rotation and translation, without altering the shape or size of the object.
- **Affine Transformation:** Extends rigid transformations to include scaling and shearing, thus preserving points, straight lines, and planes while allowing volume change.
- **Deformable Transformation:** Allows local bending and stretching of structures, adapting to more complex and variable changes that do not adhere to fixed linear transformations.

Each of these can be expressed by a fixed set of parameters. Similarly to segmentation, one can distinguish between manual, semi-automatic, and automatic techniques, differing in the level of required human interaction. Similarly to other computer vision tasks, deep learning-based approaches have achieved state-of-the-art performances recently [16, 101, 32]. Image registration has been used for medical imaging tasks, including landmark prediction [24, 13, 82]. A so-called atlas, a reference image with known voxel coordinates $p_l$ of landmark $l$, is registered to a target image with unknown voxel coordinates, obtaining a transformation $T$. Then, the landmark coordinates in the target image are obtained as $T(p_l)$. However, these methods are time-consuming and present limited robustness due to the great geometrical variations on finer scales often found in medical images [46].

# 3 Related Work

This section is dedicated to reviewing previous works relevant to this thesis. We will review recent deep-learning-based approaches for anatomical landmark prediction, analyze potential areas for improvement, and review works that have focused on learning using morphological representations instead of direct grayscale images in medical image analysis tasks.

**Deep Learning in Anatomical Landmark Prediction** Recent works in landmark prediction predominantly fall into one of two formulations: Coordinate Regression methods [93, 99, 91, 23, 69] directly model a mapping from the image to the landmark coordinates, represented as a collection of 2D- or 3D vectors. Conversely, in Heatmap Regression [86, 72, 65] learns a pixel-wise mapping from the image to an output representing a pseudo-probability distribution, where for each pixel a probability is estimated for containing each landmark. Yang *et al.* [93] propose a 2.5D approach, processing slices of MR images of femurs slice by slice, being one of the first to largely rely on a CNN for landmark detection. However, their method is not end-to-end trainable and involves hand-selected geometric features like curvature that do not generalize to all settings. Zheng *et al.* [99] developed another two-stage approach, employing two MLPs to generate candidates and extract features patch-wise. However, their method is inefficient by using an MLP for voxel-wise classification and is infeasible on large scans. It also relies on hand-crafted features and elaborate boosting mechanisms. The heatmap regression approach, first introduced by Tompson *et al.* [86], was adapted to medical images by Payer *et al.* [72], who also propose a spatial configuration module designed to incorporate the relative positioning of landmarks. Using a CNN to process the image, their method first generates local appearance maps, potentially ambiguous since the anatomical structures of different landmarks are frequently very similar on a local level. To counteract this issue, spatial configuration maps, predicted from the local maps and intended to capture the global structure, filter the appearance maps through element-wise multiplication on a per-landmark basis. In a more recent work, Gao *et al.* [23] follow a similar approach of combining a CNN based architecture with learned global structure information, limiting confounding background signals. They achieve this by a self-attention mechanism incorporated at multiple stages of a stacked hourglass network that had previously shown success in human pose estimation [65]. Their work uses the soft-argmax operator, proposed by Luvizon *et al.* [61], which offers a differentiable way to retrieve coordinates from a heatmap. This allows direct supervision using coordinates directly instead of artificially generated ground truths while learning heatmaps implicitly, leveraging CNNs inherent strength to retain spatial consistency. Both these methods have shown significant advancements in accuracy, highlighting the benefit of including global structural information. Notably, though, while both methods learn an implicit representation of global structure in that a promising region is identified for each landmark, they do not explicitly model the relationship of landmarks to each other. Nguyen *et al.* [67] proposed a method to explicitly take advantage of the global structure by modeling the landmarks in a facial landmark detection task as nodes of a graph, which are described by coordinates and features extracted from a CNN backbone and then processed with a Graph Convolutional Network (GCN), an architecture specifically designed for processing graph-like data. Inspired by the recent success of transformer architectures, in particular the DETR architecture proposed by Carion *et al.* [10] for object detection, Watchareeruetai *et al.* [91] employ a transformer architecture to model inter-landmark dependencies in a coordinate regression approach. Similarly, Li *et al.* [57] use cascaded transformers to refine coordinate predictions while capturing structural dependencies. The choice of a transformer over a GCN has a strong theoretical foundation as transformers have been shown to be highly efficient at learning on graph-based data [48]. However, while both of these works use a CNN for feature extraction, they flatten and reshape the feature map before applying the transformer module, thereby destroying the spatial feature flow of

the CNN, which has been cited as a weakness of coordinate regression compared to heatmap regression [61]. Chen *et al.* [11] propose a method combining coordinate regression and heatmap regression, using a U-Net to create landmark-wise heatmaps and an Long Short-Term Memory (LSTM), processing iteratively processing patches of increasing resolution to refine the predictions. Their approach extracts landmark-wise features based on their predicted spatial location in the feature map at the bottom layer of the U-Net. It uses a self-attention mechanism to model the global structure. Achieving state-of-the-art performance in cephalometric landmark detection, their approach showcases the potential of combining the feature representation of CNNs with inter-landmark dependency modeling as it had been explored for 2D facial landmarks by Watchareeruetai *et al.* and Ngyuen *et al.* [91, 67]. Still, it does not make use of spatial attention in the CNN backbone, which has shown improvements in the works of Payer *et al.* and Gao *et al.* [72, 23].

**Input Representation In Medical AI** While the direct processing of grayscale values as obtained by CT scanners represent the majority of works in medical image processing, several recent works have explored the use of abstractions of the image encoding the morphology of analyzed inputs. Mohammed *et al.* use segmentation masks of CT images to classify individual vertebrae based on morphology. In a subsequent work, Meng *et al.* use a binary segmentation mask of the spine for vertebra localization, segmentation, and identification [62]. Sekuboyina *et al.* use a point cloud, a set of 3D points in space, extracted from a segmentation mask to encode the shape of vertebrae to detect fractures [81]. Ankut *et al.* use semantic segmentation masks of individual vertebrae to identify and classify lumbosacral transitional vertebrae [4].

# 4 Materials and Methods

This chapter explains the used datasets in detail, the steps we undertook to ensure data integrity and prepare the data for our task, and the model we developed to solve it, explaining and justifying the design choices. We will conclude this chapter by describing the loss used for optimization and the metrics employed to evaluate our model's performance.

## 4.1 Datasets

This thesis utilizes two in-house datasets representing different clinically significant landmarks. One of the datasets contains coordinates of ligament attachment points on vertebrae; we will refer to this dataset as the Ligament Attachment Point Dataset (LiAP Dataset). The second dataset contains annotations of the expected location of pedicle screw heads and tips Both follow the brain imaging data structure (BIDS) format [27], organizing the available data per subject into raw data and derivatives.

**Raw Data** For each subject in the LiAP Dataset and Pedicle Screw Dataset (PS Dataset), a CT image of the subject, showing the partial or whole spine, is included, along with accompanying metadata.

**Derivatives** The raw CT scans have been automatically processed to generate two types of segmentation masks per scan. The instance segmentation mask segments the spine into individual vertebrae, with each vertebra labeled according to a standardized key (e.g., '1' for C1, '2' for C2, etc.). The semantic segmentation mask further divides each vertebra into ten distinct subregions, providing detailed anatomical information that can be leveraged for POI analysis. In addition to the segmentation masks, each dataset includes manual expert annotations of POIs in JSON format. These annotations represent clinically significant landmarks identified on the scans, representing the developed models' prediction targets. The following subsections detail each dataset's specific characteristics and unique aspects, including the nature of the POIs and the particular clinical scenarios they represent.

### 4.1.1 The Ligament Attachment Point Dataset

The first dataset contains the scans and derivatives explained above for 36 subjects, alongside annotations of attachment points of the six ligaments described in 2.1.2 at each vertebra. We will refer to this dataset as the LiAP Dataset. In general, the ligaments do not attach at a singular point but cover an area of contact with the vertebra. This area is particularly large for ALL and PLL, wrapping around the vertebra corpus at the anterior and posterior side, respectively, and the FL, which contacts a larger area at the vertebral arch. Thus, to allow for an accurate representation of the contact area six points were marked on each vertebra for each of the above-mentioned ligaments:

- The center point along the width of the ligament at both the cranial (superior) and caudal (inferior) extreme of the attachment area to the vertebra.
- The left-most and right-most points, each also at the cranial and caudal extreme.

Figure 4.1 shows a depiction of the ligaments and a visualization of the POIs corresponding to the ALL in the LiAP Dataset. All visualizations of the POIs are created with PyVista, developed by Sullivan *et al.* [85].

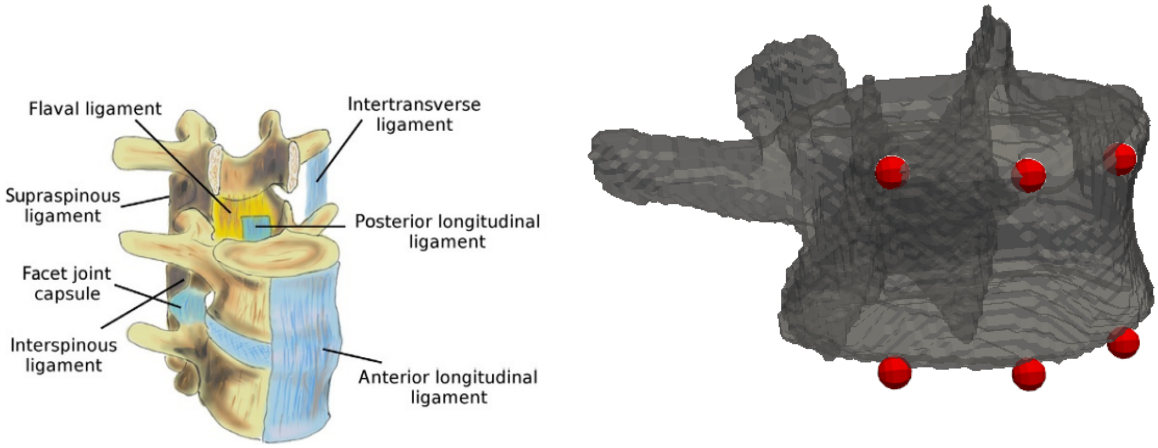



**Figure 4.1** Depiction of the ligaments of the lumbar spine (left, taken from [22, p 55]) and visualization of the annotations of the ALL in the LiAP Dataset (right). The red dots mark, beginning at the upper-left in clockwise direction: ALL cranial dexter, ALL cranial, ALL cranial sinister, ALL caudal sinister, ALL caudal, ALL caudal dexter.

The attachment area of the other ligaments, namely the interspinous, intertransverse, and supraspinous ligaments, is much smaller and may be reasonably approximated by a single point.

For the interspinous ligament, there are two annotated attachment points per vertebra: The cranial attachment point, which marks the attachment point of the segment connecting the corresponding vertebra to its cranial neighbor and analogously, the caudal attachment point. The intertranverse ligaments attach at the outside of the processus transversus on each side of both vertebrae, hence the left and right attachment point are marked independently for each vertebra. The supraspinous ligament attaches at a single point on each vertebra at the posterior extreme of the processus spinosus, and is hence marked exactly once for each vertebra. Table 4.1 shows an overview of all POIs present in the LiAP Dataset with corresponding abbreviations which we will adopt in this thesis.

| Ligament | Height | Left | Middle | Right |
|---|---|---|---|---|
| Anterior Longitudinal Ligament (ALL) | cranial | ALL_CR_S | ALL_CR | ALL_CR_D |
| | caudal | ALL_CA_S | ALL_CA | ALL_CA_D |
| Posterior Longitudinal Ligament (PLL) | cranial | PLL_CR_S | PLL_CR | PLL_CR_D |
| | caudal | PLL_CA_S | PLL_CA | PLL_CA_D |
| Flaval Ligament / Ligamentum Flavum (FL) | cranial | FL_CR_S | FL_CR | FL_CR_D |
| | caudal | FL_CA_S | FL_CA | FL_CA_D |
| Intraspinous Ligament (ISL) | cranial | | ISL_CR | |
| | caudal | | ISL_CA | |
| Intertransverse Ligament (ITL) | | ITL_CR_S | | ITL_CR_D |
| Supraspinous Ligament (SSL) | | | SSL | |

**Table 4.1** All POIs in the LiAP Dataset with their corresponding abbreviations. Letters before the underscore abbreviate the corresponding ligament, CR stands for cranial, CA for caudal, S for sinister (left) and D for dexter (right)

### Additionaly Available Derivatives

In addition to the already mentioned derivatives, namely the vertebra and subregion segmentation masks and the POI annotations, for each vertebra POIs were derived automatically, using registration as described in Section 2.4. Six subjects served as references for the registration, for each subject, the registration POIs were calculated, except for the reference subject to itself.

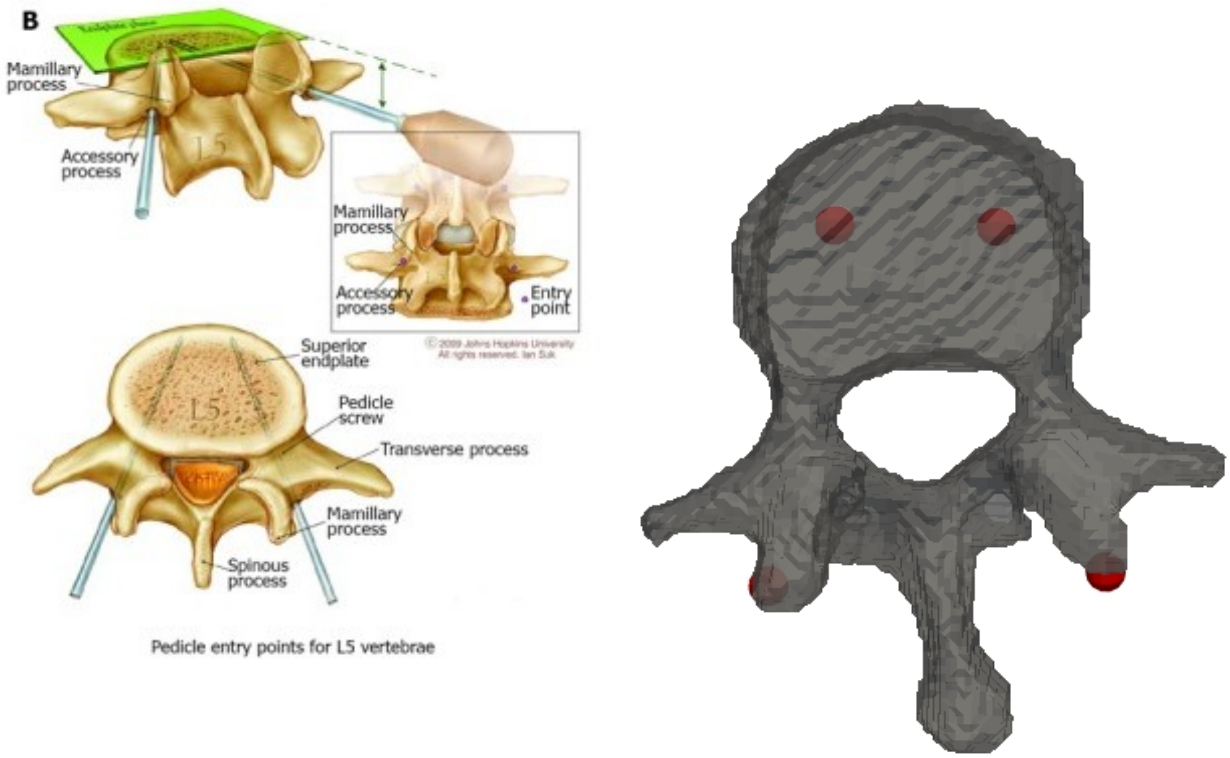



**Figure 4.2** Depiction of accurate pedicle screw placement (left, taken from [75]) and annotated screw heads and tips for an L1 vertebra in the PS Dataset. (Created with PyVista [85]).

### 4.1.2 The Pedicle Screw Dataset

The Points of Interest annotated in the PS Dataset represent the expected locations of screw heads and tips after applying pedicle screws (see Section 2.1.3). The dataset consists of the annotations of 8 subjects with differing numbers and types of vertebrae that are annotated for pedicle screw placement, depending on the clinical indication of the specific subject. Figure 4.2 shows a depiction of the theoretically ideal placement of pedicle screws and a corresponding sample in the PS Dataset.

## 4.2 Data Analysis and Cleaning

Before our model's training and validation phases, it is crucial to thoroughly examine the dataset to identify outliers and verify the realism of the data. This eliminates potentially erroneous annotations and facilitates informed decision-making in subsequent processing steps. This section offers a detailed overview of the composition of the datasets and the exploratory data analysis conducted. Furthermore, we describe the data cleaning and preprocessing methods implemented to guarantee that the model receives clean, uniformly formatted data. These preliminary steps are essential for laying a solid foundation for the robust performance of our automated point-of-interest prediction in CT scans of human vertebrae.

### 4.2.1 Ligament Attachment Point Dataset

A fundamental property of the POIs present in this dataset is that, as attachment points for ligaments to the vertebrae, they must necessarily reside on the vertebral surface. Moreover, for certain POI types, it is possible to specify the exact subregion of the vertebra where they are expected to be located. For example, the ALL, which encircles the anterior portion of the vertebral body, dictates that all its associated attachment points are located on the surface of the vertebral body. Similar constraints apply to other

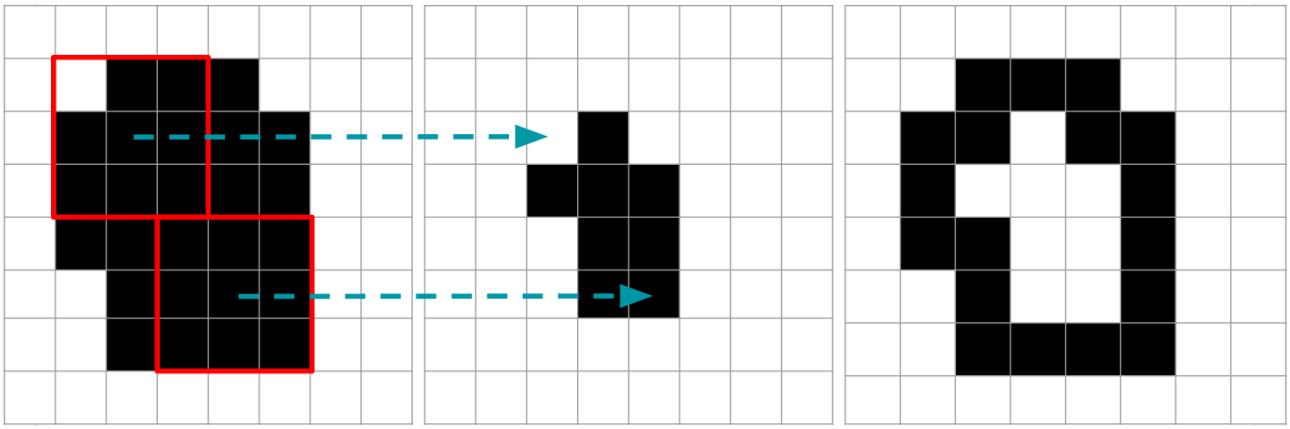


**Figure 4.3** A 2D Example of surface computation using binary erosion with a three-by-three structuring element. The pictures shows an input (left) with example positions for the applied erosion, the resulting eroded input (middle) and the surface as the difference between original input and erosion (right).

POI types. Given that the dataset includes accurate subregion segmentations, we can ensure that these criteria are met in an automated fashion, as detailed in this section.

A two-step computation delineated the surface from the vertebra segmentation mask. The initial step was to compute the inner points of the vertebra using binary erosion [34]. Following the binary erosion, the vertebra surface is computed by subtracting the eroded inner points from the original segmentation mask. This subtraction highlights the boundary layer of the vertebra, effectively outlining the surface. Figure 4.3 graphically demonstrates this process using a 2D representation for clarity.

The next step involves computing the Euclidean distance from each POI to the nearest surface point, identifying the closest surface point using the nearest neighbor approach. Additionally, we utilize the index of this nearest neighbor within the subregion segmentation mask to determine the closest subregion to the POI. Figure 4.4 shows the distribution of the distances to the surface determined by this method. We can see that the majority of POIs roughly follow a normal distribution against the logarithm of the surface distance However, the distribution is strongly tail-heavy, with a significant portion of surface distances exceeding 10mm. The median of all 19734 POIs was calculated to be 3.28mm with a standard deviation of 17.28mm and a mean of 1.02mm, indicating the high amount of outliers on the high side.

To broaden the understanding of the underlying data, the mean and median were also calculated on a per-subject, per-poi-type and per-vertebra-type basis. By far the biggest variations were seen in the per-subject statistics, indicating that the issue is likely to be corrupted data of individual subjects. Figure 4.4 shows the cumulative distribution of the mean distances computed on a per-subject basis. Using the common elbow point heuristic, it was decided to inspect the eight subjects with the highest mean distances visually. This inspection revealed that the data showed inconsistencies on a global scale for five of the given subjects, likely stemming from misaligned segmentations being used during annotation, while a further three subjects contained a significant amount of outliers, often concentrated at a few vertebrae, affecting the overall statistics. In contrast, the majority of annotations are still usable.

The surface distance represents only one of the measures that can be analyzed automatically. Another one is given by determining which subregion the nearest neighbor of a POI among the surface points belongs to, according to the subregion segmentation mask. Combined with the surface distance, this represents a strong indicator of plausibility for the annotation: If a POI is shifted away from the centroid of the vertebra, the surface distance increases, but the closest subregion may still be the appropriate one. Conversely, if a POI is shifted on or close to the surface, a sufficiently large shift will result in the POI being closer to a different subregion than intended. In particular, the method of identifying the closest subregion can help identify mislabelling, i.e., cases where the marked coordinates do belong to one of the Point of Interest, but not the one that it is labeled as.

Based on the anatomical properties of the ligaments described in 2.1.2, in particular the attachment sites to the vertebrae, for each POI one or two "acceptable" subregions were identified. In 17 cases, the

expected subregion can be clearly identified and is the only acceptable subregion; any other indicates an annotation conflicting with the anatomical properties. In the remaining six cases, two subregions are possible for the attachment point to lie on, e.g. the outer attachment points of the PLL may lie either on the vertebra corpus border or the arcus, depending on the width of the ligament. The analysis of the closest subregions shows that overall, 94.1% of all POIs are closest to one of the subregions they are expected on, while the remaining 5.9% are not.

A further test we can conduct automatically based on prior knowledge of the anatomical properties is a check for spatial consistency. Considering a group of POIs associated with the same ligament, e.g., the six attachment points of the ALL, we verify whether the order in the left-right direction and the order in the superior-posterior direction are respected by the annotations, i.e. we verify whether it holds that ALL_CR_S is annotated to the left of ALL_CR and this again to the left of ALL_CR_D and that each of these POIs are annotated above their caudal counterpart.

Based on these findings, the dataset underwent a rigorous cleaning process to balance enforcing stringent quality standards and retaining a substantial volume of data. This balance was essential to ensure the integrity of the data while avoiding excessive reduction in dataset size. To achieve this, four heuristic criteria were implemented:

- The five subjects exhibiting global inconsistencies were dropped entirely.
- Vertebrae with a mean surface distance of more than 2mm among all POIs were dropped.
- Vertebrae where the POIs did not adhere to all spatial consistency criteria were dropped.
- Individual POIs with a surface distance of more than 3mm were marked 'missing'
- Individual POIs whose nearest neighbor on the surface did not belong to an acceptable subregion were marked 'missing'

Tables 4.2 details how many of the vertebrae were accepted and dismissed after the removal of 5 of the overall 36 subjects due to the failed visual inspection, separated by the reasons, and Table 4.3 shows how many of the POIs were accepted and dismissed after additionally cleaning on the vertebra-level.

| | **Spatially Consistent** | **Not Spatially Consistent** | **Total** |
|---|---|---|---|
| **Acceptable Mean Projection Distance** | 547 | 25 | 572 |
| **Not Acceptable Mean Projection Distance** | 102 | 4 | 106 |
| **Total** | 659 | 29 | 698 |

**Table 4.2** Data cleaning results on a vertebra level, after removing subjects that failed the visual inspection. 547 of 698 vertebrae met both criteria and were retained in the dataset.

| | **Acceptable Subregion** | **Not Acceptable Subregion** | **Total** |
|---|---|---|---|
| **Acceptable Projection Distance** | 12,295 | 136 | 12,431 |
| **Not Acceptable Projection Distance** | 144 | 6 | 150 |
| **Total** | 12,439 | 142 | 12,581 |

**Table 4.3** Data cleaning results on a POI level, after removing subjects that failed visual inspection and cleaning on a vertebra level. 12,295 of 12,581 POIs were retained and used for training.

### 4.2.2 The Pedicle Screw Dataset

We verified the spatial configuration's consistency similarly to the analysis on the LiAP Dataset. We confirmed that the coordinates of POIs labeled as left head and tips are true to the left of their right counterparts and that POIs labeled as tips are anterior to the heads, complying with the definition. Since the

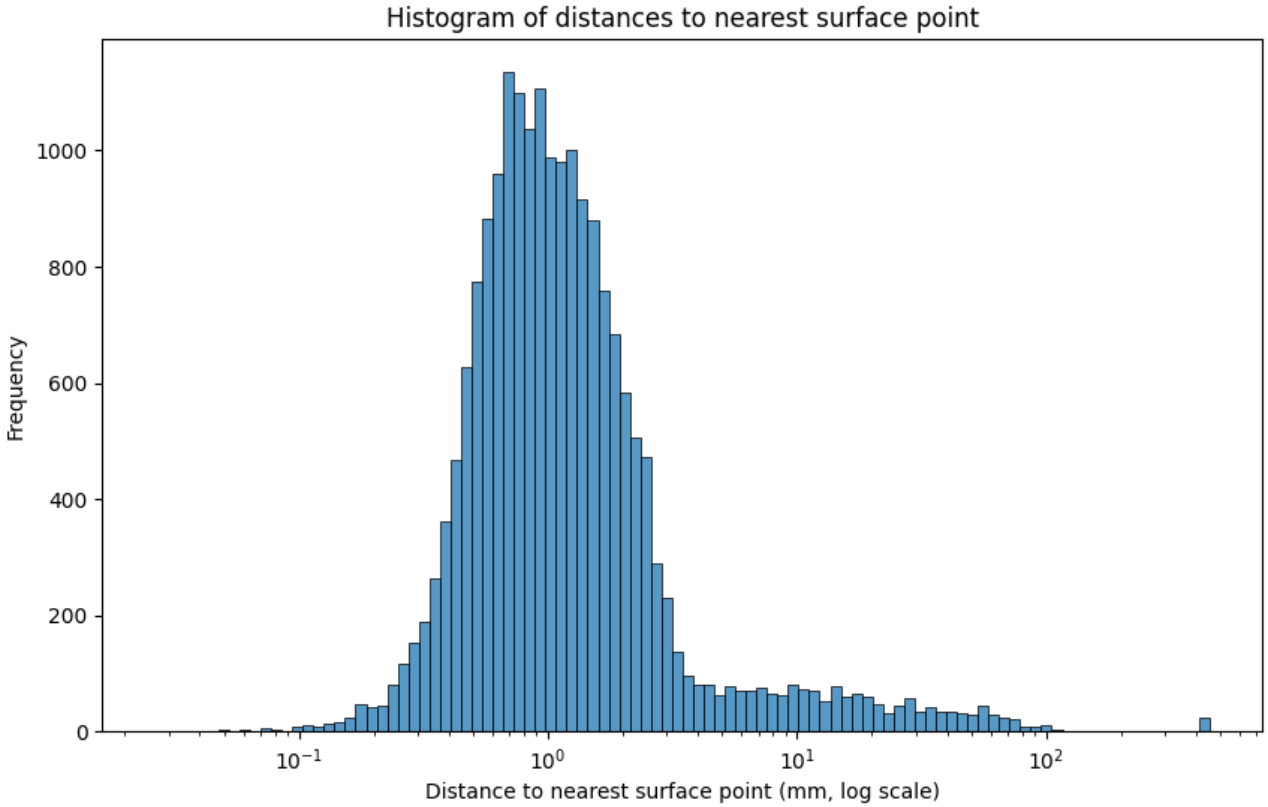



**Figure 4.4** Distribution of the distances to the nearest point on the surface of the corresponding vertebra among all POIs in the LiAP Dataset dataset.

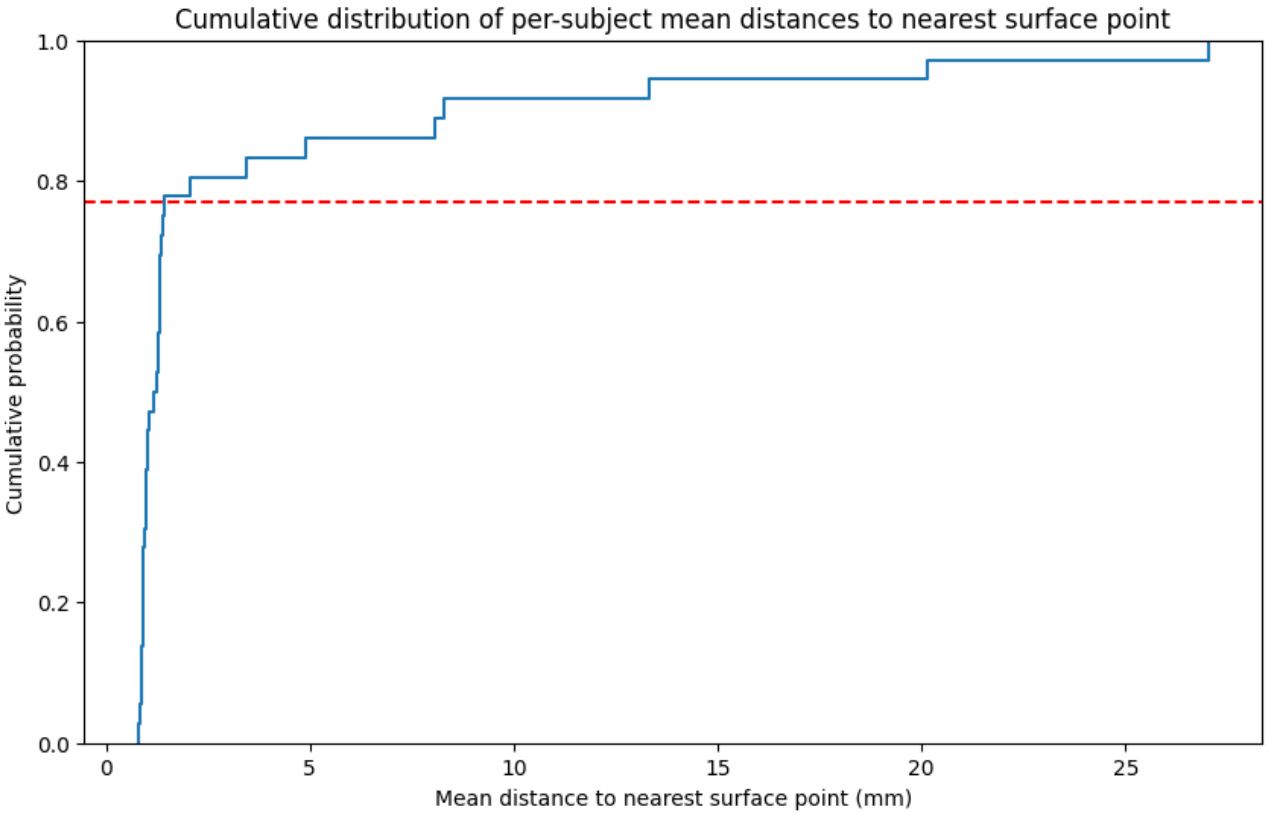



**Figure 4.5** Cumulative distribution of per-subject mean distances to the nearest surface point. The 8 samples with the highest mean surface distance (above the red line) were manually inspected.

screw heads and tips annotated in the PS Dataset are not expected to lie on the surface, the analysis techniques based on surface projection distance from above can not be applied. However, the small size of the dataset, consisting of only 8 subjects, allows for visualization of all samples. This revealed a higher prevalence of pathological cases within this dataset, making the pedicle screw treatment necessary. We do, however, not have expert annotations to quantify this phenomenon.

## 4.3 Data Preprocessing

To achieve a consistent and reliable data format, the datasets were preprocessed in the following way: Using the vertebra segmentation masks, bounding boxes were generated for each vertebra, and the vertebra was cut accordingly. The semantic segmentation mask was cut out using this bounding box, and the locations of the POI were shifted to match the cropped version. In the next step, the cutout segmentation masks and the POIs were rescaled to 1mm isotropic voxel resolution, and finally, the segmentation masks and POIs were reoriented to a common orientation.

Given that quality assessment of pedicle screw placement focuses primarily on the entry point as well as the trajectory within the narrowest bone corridor, the pedicle, and further, the exact depth of the screw is less critical than the placement within it as explained in Section 2.1.3, we did not directly use the annotated head and tip points. Instead, we calculated the entry points of the screw into the vertebra as a whole and into the vertebral body, i.e., the point on the screw trajectory that sits at the junction of the arcus and the vertebral body. These mark the points just before and after the critical pedicle, providing more reliable prediction candidates while fully defining the line on which the trajectory lies. In a clinical scenario, the coordinates of the tip and head of the screw can easily be retrieved from these trajectory key points by shifting the points along the connecting line to the desired depth of the screw.

To obtain the specified points, the trajectory of each screw was discretized into 100 equidistant steps from the head coordinate $P_H$ to the tip location $P_T$, resulting in the screw sample points $P_i = \frac{i}{100}(P_T - P_H)$. For the entry point to the vertebra, we chose the $P_i$ with the lowest $i$, such that the nearest neighbor voxel belonged to any subregion in the vertebra semantic segmentation mask; the same technique was used for defining the entry point to the vertebral body by considering only voxels belonging to the vertebral body in the segmentation mask. Figure 4.6 compares the annotations before and after computing these key points.

The 31 subjects of the LiAP Dataset were split into 23/4/4 for training validation and testing. It was ensured that the validation and test set did not contain subjects with more than 20% missing annotations to ensure reliable results despite the small sample size; otherwise, the choice was random. The PS Dataset was split into 6/1/1 on a subject level and it was ensured that all vertebrae that were present in the dataset were visible on the validation and test subject.

## 4.4 Model Architecture

In this work, we consider multiple model architectures for the given POI prediction task and explore their strengths and weaknesses from a theoretical and experimental point of view. Following previous works, we tackle the task in a two-step fashion, starting with a coarse prediction and global feature extraction and refining the predictions in a second step. This two-tiered approach has several benefits: The global extraction module can work on a low-resolution version of the input, saving computational cost and potentially speeding up convergence since a coarse initial estimate suffices if the refinement module is powerful enough. Conversely, the refinement module can include high-resolution local details, requiring the processing of only small parts of the overall image thanks to the coarse estimates. Moreover, we explore a refinement module designed to implicitly learn inter-landmark dependencies motivated by the highly structured geometry of the task.

Despite the recent dominance of heatmap-based approaches, we follow Li *et al.* [57] in framing the task as a coordinate regression problem, which simplifies the task in several aspects: There is no need

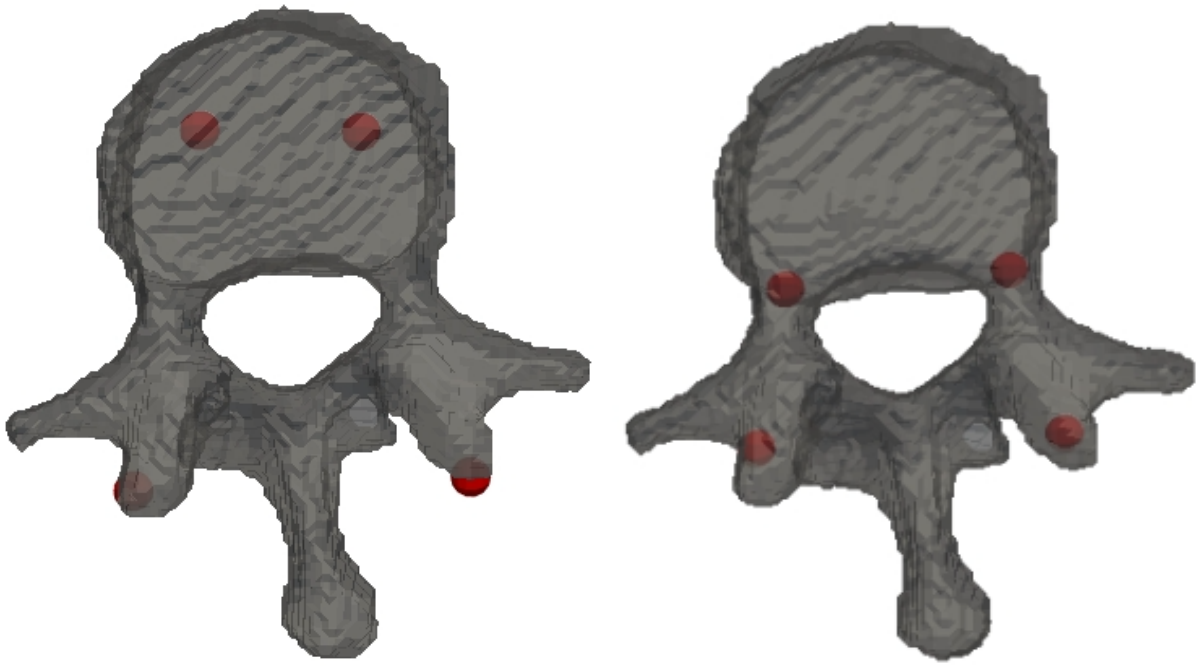


**Figure 4.6** Original annotations in the PS Dataset (left) and the trajectory key points before and after the pedicle used for training (right).

for creating artificial ground truth heatmaps, saving computational cost, the output is directly aligned with target metrics like Euclidian distance and we eliminate the sensitivity of loss terms to resolution, which is particularly useful in our multi-resolution setting. We can still integrate the aspect of maintaining the spatial nature of the data, which is a commonly cited strength of heatmap regression, by using the soft-argmax operator [61].

With these thoughts in mind, we view all explored architectures as instances of the following simple yet modular and adaptive framework (see Figure 4.7) :

1. A *feature extraction module* processes input images to produce initial landmark coordinate predictions and globally informed visual features on a per-landmark basis.

2. A *refinement module* enhances these initial predictions by facilitating feature communication across landmarks. This approach allows the model to learn geometrical relationships between landmarks implicitly.

In the proposed thesis architecture, spatial and self-attention mechanisms are utilized to refine the feature extraction and representation learning processes in CT scans of human vertebrae. Spatial attention is employed to selectively aggregate features from relevant parts of a feature map generated by a CNN, focusing on critical anatomical regions that contain crucial landmarks. Concurrently, a transformer module equipped with self-attention processes a list of rich feature vectors, each representing visual features, encoded anatomical information, and coarse coordinates of landmarks. The output of this module is a refined set of features, where each feature vector has been adjusted to more accurately predict the positional offset from the initial coarse coordinate, leveraging the self-attention's ability to integrate contextual information from the entire set of landmarks. Thus, incorporating these attention mechanisms not only harnesses the power of spatial and contextual relevance but also aligns with the overarching goal of achieving precise and robust automated point-of-interest prediction in medical imaging.

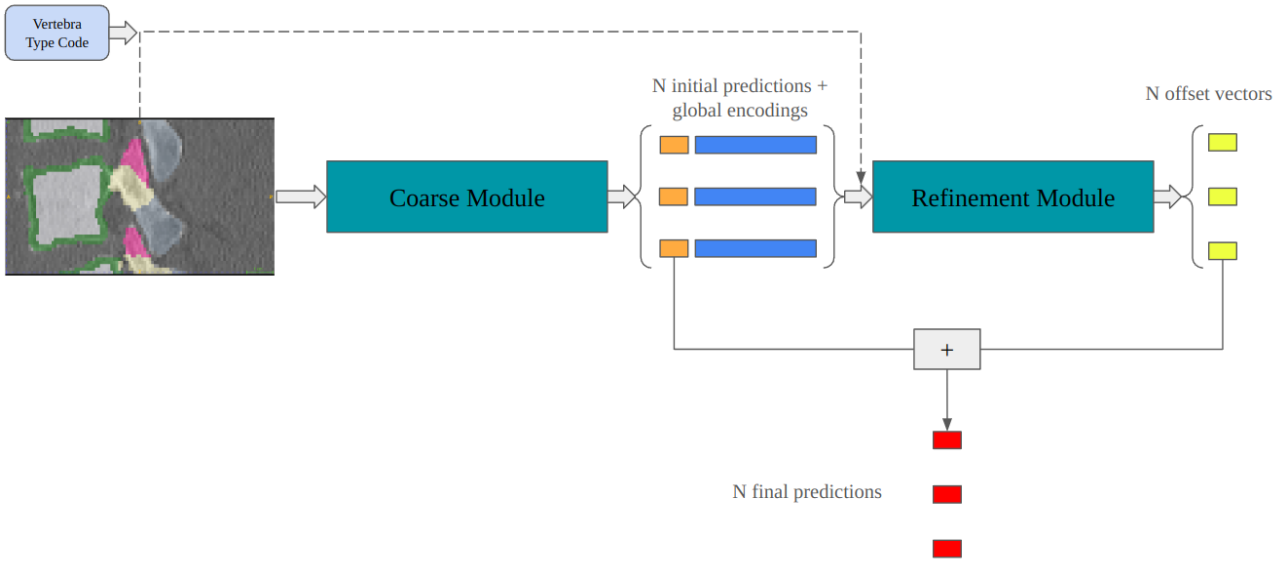



**Figure 4.7** The framework used for the task.

### 4.4.1 DenseNet-Based Feature Extraction and Coarse Estimation

We employ a feature extraction module based on the classical DenseNet architecture described in section 2.2.2. To adapt the architecture for our purposes, however, we must extract features on a per-POI level and produce the initial coarse coordinate estimate for each landmark. A straightforward idea to implement this would be to remove the final soft-max layer and predict, for $N$ acppoi with a feature encoding length of $F_G$, a feature vector of length $N * (3 + F_G)$. This output can then be reshaped and split into a landmark-wise coordinate representation of shape $(N, 3)$, and the landmark-wise features of shape $(N, F_G)$.

However, motivated by the findings of works that have employed heatmap regression techniques over coordinate regression, we believe that this approach is not ideal: The commonly cited advantage of heatmap regression is that convolutional layers inherently respect the spatial configuration of the image, i.e., even in deep feature maps there is still the notion of a spatial relationship between feature vectors which corresponds to the relationship of patches in the original image. This spatial relationship, however, is destroyed when flattening and pooling operations are applied. Therefore, we propose a different approach, as depicted in Figure 4.8.

The final pooling and flattening layers are omitted, resulting in a deep feature map with a spatial extension, respecting the original image but downsized by DenseNet's transitioning layers. A final convolutional layer simultaneously produces landmark-wise heatmaps and a deep feature map, represented as a tensor of shape $(N + F_G, H, W, D)$. The first $N$ (along the channel dimension) maps are interpreted as heatmaps for the localization of the landmarks. Through a soft argmax, we can retrieve coarse coordinates from here and supervise these maps with the ground truth coordinates. Then, the softmax-normalized maps are applied via component-wise multiplication and summation across the spatial dimensions to produce the $N$ feature vectors of size $F_G$. In this sense, the heatmaps can be viewed as attention maps, aggregating for each POI the features from the deep feature map corresponding to its spatial location while suppressing the background. By employing a sufficiently large number of convolutional layers in generating the heatmap, the features are still globally informed, in the sense that the receptive field of each spatial location in the feature map is the entire image.

### 4.4.2 Refinement Module

While the coarse module already provides estimates for the landmark coordinates, we explore a refinement module, potentially enhancing the initial predictions. Inspired by the tremendous success of the

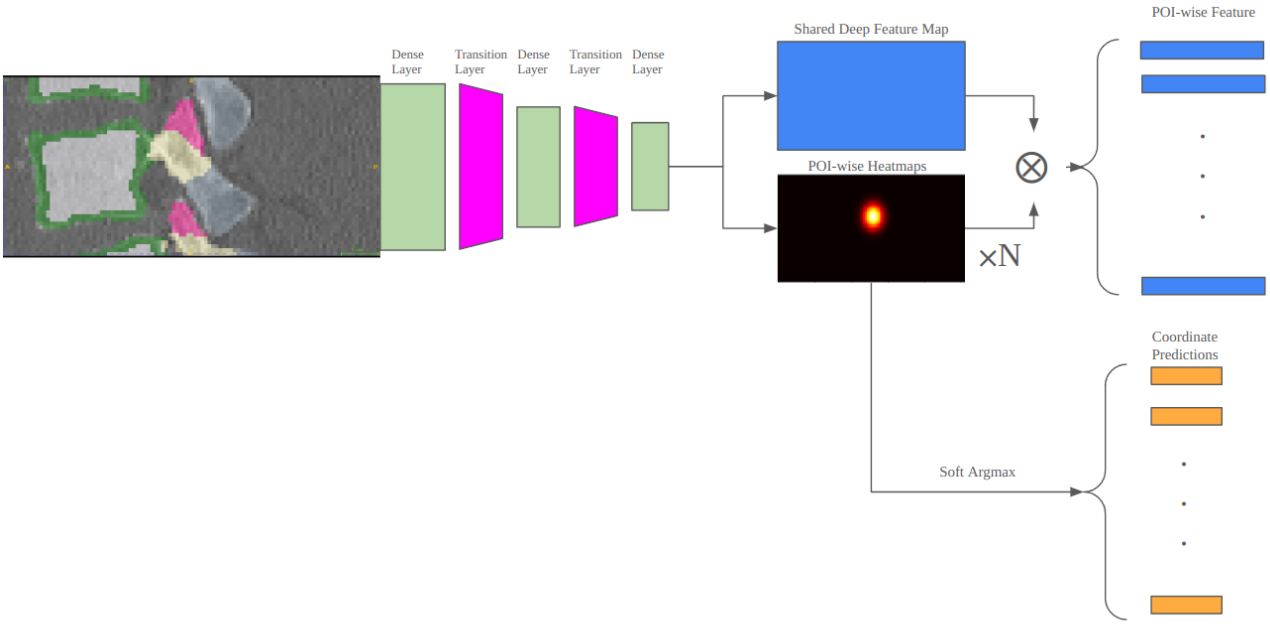



**Figure 4.8** The proposed Coarse Extraction Module.

transformer architecture in related vision tasks and the inherent capability of the transformer to learn relationships between the inputs, we decided to employ a transformer architecture. The idea behind this design choice is the following: The POIs we are considering exhibit a high degree of regularity in terms of their relations to each other, e.g., the spatial configuration in an idealized example is symmetric, widths of specific ligaments, represented by their POIs are related to each other, specific POI are coplanar or lie on a curved plane in space. By learning to relate the POIs to each other, we conjecture that the transformer can help maintain spatial coherence and improve the coarse predictions by correcting inconsistencies. Our refinement module is strongly related to both the UNETR and ViT architectures discussed in Section 2.2.3: Both these architectures rely on feeding patch-wise encodings to a transformer encoder and use the output for a downstream task. However, differing from both these approaches and saving computational costs, we do not consume the entire image at this point but instead, pre-selected patches, which we encode by with a small Convolutional Neural Network into the Patch Feature encoding $F_P$. We include encodings of the vertebra and POI type for each POI, the patches' center coordinates, and the POI-wise global features $F_G$ extracted with the coarse module. The reasoning behind the incorporation of each of these features can be summarized as follows:

- **Vertebra and POI type encoding $F_V$ and $F_P$**: Includes high-level semantic information and allows the transformer to adapt the processing under consideration of the vertebra and POI type. Some landmarks may interact differently with each other based on the vertebra type, e.g., landmarks on a thoracal vertebra exhibit a different geometrical configuration than those on a lumbar vertebra, and even within a spine region, there is some variation in morphology as discussed in Section 2.1.1.
- **Patch Center Encoding $F_C$**: Similar to a positional encoding in existing transformer architectures, the patch center encoding provides spatial context within the image. Since the location of the patches varies in each iteration, depending on the coarse module's predictions, we believe this encoding to be crucial for accurately relating the patch features.
- **POI-wise Global Features** include global context, allowing the model to capture specific morphological features that may impact POI positionings but are not evident in the smaller patches.

Figure 4.9 depicts an overview of the entire architecture.

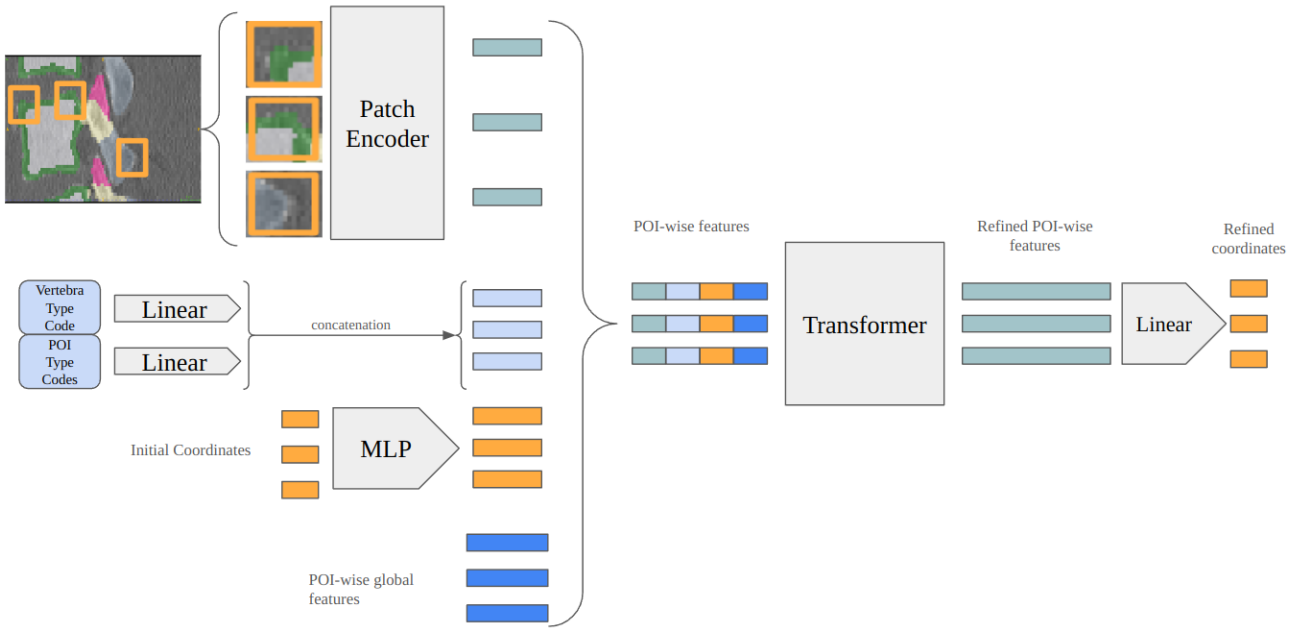



**Figure 4.9** The Proposed Transformer-Based Refinement Module.

## 4.5 Loss Function and Metrics

All models are trained using a 3D version of wing loss. Wing loss, first introduced by Feng *et al.* [19], is a loss function initially designed for the related task of facial landmark detection.

Mathematically, Wing Loss is defined piece-wise

$$\text{Wing}(y, \hat{y}) = \begin{cases} w \ln(1 + \frac{|y-\hat{y}|}{\epsilon}) & \text{if } |y - \hat{y}| < w, \\ |y - \hat{y}| - C & \text{otherwise,} \end{cases} \tag{4.1}$$

where $y$ is the ground truth value, $\hat{y}$ is the predicted value, $w$ defines the width of the non-linear part, i.e., the range of small and medium errors that are supposed to be emphasized, and $\epsilon$ controls the curvature of the non-linear part of the loss function $C$ is a constant given by $C = w - w \ln(1 + \frac{w}{\epsilon})$ to ensure continuity of the loss function. Figure 4.10 plots the loss function for a fixed $w$ and different shape parameters $\epsilon$.

The primary motivation behind Wing loss is to address the insensitivity of traditional loss functions like L1 and L2 to minor errors. L2 loss, for instance, squares the differences between predicted and actual values, which tends to disproportionately penalize more significant errors while being less sensitive to more minor discrepancies. On the other hand, L1 loss, which takes the absolute difference, provides equal weighting to all errors but can lead to less stable gradient updates due to its linear nature.

Wing loss introduces a logarithmic curve for errors below a certain threshold (determined by the parameter $w$), making it more sensitive to minor errors than L1 and L2 and less sensitive to outliers. Both aspects are promising in our context: Sensitivity to minor errors ensures a satisfactory performance and can help the model consider finer anatomical details in the high-resolution image patches processed during the refinement stage. On the other hand, judging by the discoveries in the data cleaning steps, we suspect there may still be a significant amount of mislabelled or otherwise inaccurate annotations that we could not identify with the automated data cleaning steps. With losses like L2, such inaccurate annotations with large deviations have a much more significant impact than correct annotations where the prediction error is presumably much smaller.

Following the original implementation, we choose $\omega = 5$ and $\epsilon = 2$. We calculate the loss based on the coarse and refined predictions and use their sum as the final loss to optimize. We mask the loss to handle

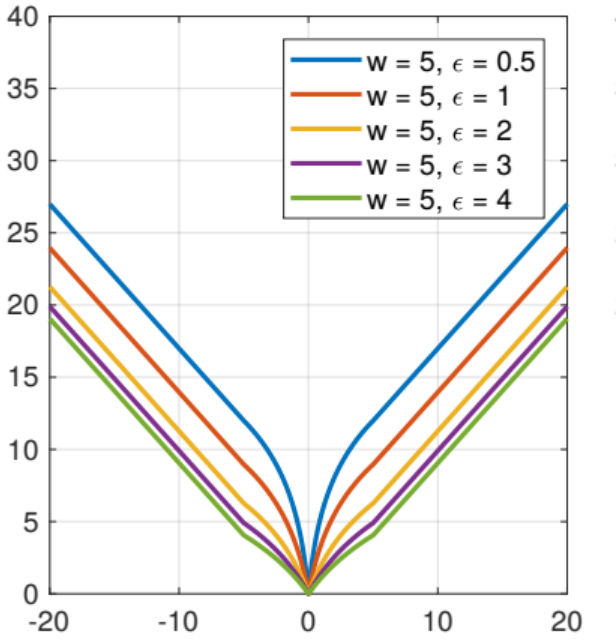



**Figure 4.10** Wing loss function plotted with $w = 5$ and different curvature parameters $\epsilon$. The x-axis indicates the L1-distance between ground truth and prediction, the y-axis the resulting loss.

the missing ground truths resulting from our data-cleaning steps so that only the remaining annotations are considered.

To assess the performance and compare different models, we report the following metrics:

**Mean Error:** The average Euclidean distance between the true landmarks ($y$) and the predicted landmarks ($\hat{y}$). It is calculated as:

$$\text{Mean Error} = \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\| \tag{4.2}$$

This metric provides a straightforward measure of the average prediction accuracy across all landmarks, indicating overall model precision.

**Median Error:** The median of the Euclidean distances between the true and predicted landmark positions across all landmarks:

$$\text{Median Error} = \text{median}(\|y_i - \hat{y}_i\|) \tag{4.3}$$

for $i = 1, 2, \ldots, N$. The median error is less sensitive to outliers than the mean error, providing a more robust measure of central tendency.

**Mean Squared Error (MSE):** The average of the squares of the Euclidean distances between the true and predicted landmark positions:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2 \tag{4.4}$$

Due to squaring the distance, MSE helps emphasize more significant errors than smaller ones, which can help detect a tendency for significant deviations.

**Accuracy:** The proportion of landmarks that are correctly predicted within a threshold distance from their true positions:

$$\text{Accuracy} = \frac{\text{Number of correctly predicted landmarks within 2mm}}{N} \tag{4.5}$$

Accuracy is instrumental as a straightforward indicator of the model's effectiveness in locating landmarks within a clinically acceptable error margin. We adopt the 2mm threshold from the related task of cephalometric landmark detection, where it is commonly deemed clinically acceptable [89].

# 5 Experiments and Results

In this section, we present the experiments we conducted and obtained results, aiming to understand the presented model's capability at solving POI prediction. We compare different configurations on the LiAP Dataset dataset to gain insight into the effects and contributions. We chose the dataset because it is significantly larger and has a significantly higher number of POIs per sample, providing more reliable performance estimates than the PS Dataset dataset. After choosing an appropriate model, we explore how well it translates to the PS Dataset and if we can leverage techniques from semi-supervised learning settings, namely masked loss, and self-training, to benefit from the close relation of the two POI types and make the best use of the data we had to remove due to being implausible.

## 5.1 Ligament Attachment Point Dataset

The LiAP Dataset dataset is the larger of our datasets and has a significantly higher number of annotations per sample. Therefore, it was mostly used to assess the performance of our model. This section presents the experiments conducted, highlighting the motivation behind different approaches to gain insight into the model's performance.

### 5.1.1 Baseline Analysis

The derivatives generated based on registration form our first baseline. We averaged the predictions generated over the different reference subjects and evaluated the performance on the validation and test set. The ground truth was cleaned beforehand following the steps outlined in Section 4.2. The same idea that motivates data cleaning based on the surface distance can also be used in post-processing: Knowing that the POIs in this dataset must lie on the surface of vertebrae, we project the ground truth as well as the prediction to their nearest neighbor voxel on the extracted surface. Table 5.1 summarizes the metrics achieved on the validation and test set.

| Split | Method | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | No Projection | 9.50 | 2.86 | 63.34 | 0.31 |
| | Surface Projection | **3.88** | **2.45** | **7.12** | **0.37** |
| Test | No Projection | 3.68 | 3.14 | 4.40 | 0.25 |
| | Surface Projection | **3.15** | **2.45** | **3.88** | **0.34** |

**Table 5.1** Registration-Based Metrics on the LiAP Dataset Dataset.

Noticing the extremely high mean and Mean Squared Error (MSE) in the validation set compared to the median before projecting the ground truths to the surface, we remove the predictions exceeding the 95th percentile of distances to the nearest neighbor on the surface to obtain the results in Table 5.2. Since this step does not use the ground truth, it can be easily integrated into real-world applications based on a pre-defined threshold to avoid implausible results.

### 5.1.2 Hyperparameters and Training

After establishing the baseline, we train our model in different variants. This section outlines the hyperparameters that were chosen for all experiments.

| Split | Method | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | No Projection | 3.33 | 2.70 | 4.30 | 0.32 |
| | Projection | **3.07** | **2.45** | **4.18** | **0.38** |
| Test | No Projection | 3.48 | 3.03 | 4.10 | 0.26 |
| | Projection | **3.12** | **2.45** | **3.84** | **0.34** |

**Table 5.2** Registration-based metrics on the LiAP Dataset after removing outliers based on surface projection distance.

**Feature Encoding Size** Table 5.3 shows the encoding length of the features that the refinement module takes in.

| **Explanation** | **Hyperparameter** | **Value** |
|---|---|---|
| POI-wise Global Encoding | $F_G$ | 256 |
| Vertebra Type Encoding | $F_V$ | 128 |
| Patch Encoding | $F_P$ | 64 |
| Coordinate Encoding | $F_C$ | 64 |

**Table 5.3** Length of the Feature Encodings.

**Optimization Parameters** We train all model versions with the AdamW optimizer [60] and the hyperparameters shown in Table 5.4. We use a linear learning rate scheduler that decreases the learning rate to a factor of 0.1 of the initial learning rate during the first 20 epochs to accelerate training initially while avoiding instability in later stages. Further, we accumulate the gradients for two batches before applying backpropagation to provide more stable weight updates. The coarse and fine losses are summed with equal weights for the final loss.

| **Hyperparameter** | **Value** |
|---|---|
| Batch size | 6 |
| Gradient Accumulation Batches | 2 |
| Loss Weight Factors | (1, 1) |
| Initial Learning Rate | 0.001 |
| Betas | (0.9, 0.999) |
| Weight Decay Rate | 0.01 |

**Table 5.4** Model Hyperparameters

**Configuration of the Modules** The DenseNet that serves as the backbone in our coarse module was configured with the parameters listed in Table 5.5. Except for the block configuration, this follows the hyperparameters used in the original proposal [39]. We employ only three blocks, maintaining a higher spatial resolution in the resulting heat maps.

The transformer in the refinement module features two layers with four attention heads and a hidden size of the MLP of 512. Overall, this results in a number of 9 million trainable parameters of the entire architecture.

**Data Augmentations** We used random affine data augmentation in all experiments with the parameter detailed in Table 5.6.

| Hyperparameter | Value |
|---|---|
| Growth Rate | 32 |
| Block Configuration | [6, 12, 12] |
| Number of Initial Features | 64 |
| Bottleneck Size | 4 |
| Compression Factor | 0.5 |

**Table 5.5** Configuration of DenseNet Parameters

| Hyperparameter | Value |
|---|---|
| Rotation Range | (-20°, 20°) |
| Shearing Range | (-0.1, 0.1) |
| Translation Range | (-5, 5) |
| Scale Range | (0.8, 1.1) |

**Table 5.6** Data Augmentation Parameters

### 5.1.3 Training On the Ligament Attachment Point Dataset

In an initial approach, we trained the model on the LiAP Dataset. We excluded the C1 and C2 vertebrae since their highly specialized morphology differs greatly even from other cervical vertebrae, and some of the ligaments are not present in these vertebrae (see Section 2.1.1). Further, encouraged by the same motivation as during data cleaning and the results seen in registration, we project both the ground truth to the nearest neighbor in a pre-processing step and the prediction of the model in a post-processing step. In Table 5.7, we report the coarse and fine prediction performance for comparison on the validation and test datasets. The better metric within the same dataset is marked in bold.

| Split | Prediction Type | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | Coarse | 2.45 | 2.24 | 8.80 | 0.41 |
| | Fine | **2.36** | **2.00** | **8.43** | **0.46** |
| Test | Coarse | 2.57 | 2.24 | **9.91** | 0.38 |
| | Fine | **2.54** | **2.24** | 9.96 | **0.41** |

**Table 5.7** Comparison of coarse and fine predictions after initial training on the LiAP Dataset.

#### Analysis of Worst Cases

Intrigued by the high maximum errors and MSE values, particularly on the test set, we analyze the worst-case scenarios to understand the reasons. Similar to our investigation in data cleaning, where we analyzed the Cumulative Distribution Function (CDF) of the surface distances, we plot the CDF of the errors between the fine predictions and the ground truth and identify the elbow of the curve in Figure 5.1.

Both in the validation and test set, we see a significant number of outliers on the upper side. To gain more insight into the causes of the outliers, we aggregate the POIs into four groups which show different characteristics: The Outer group consists of the POIs marking the lateral borders of one of the ligaments ALL, PLL and FL on the cranial or caudal level, the Inner group of their counterparts marking the middle point. ITL refers to the POIs associated with the intertransverse ligament and Spinous the POIs on the spinous process, comprised of the interspinous ligament and supraspinous ligament attachment points. Table 5.8 summarizes the distribution of the outliers to the different groups.

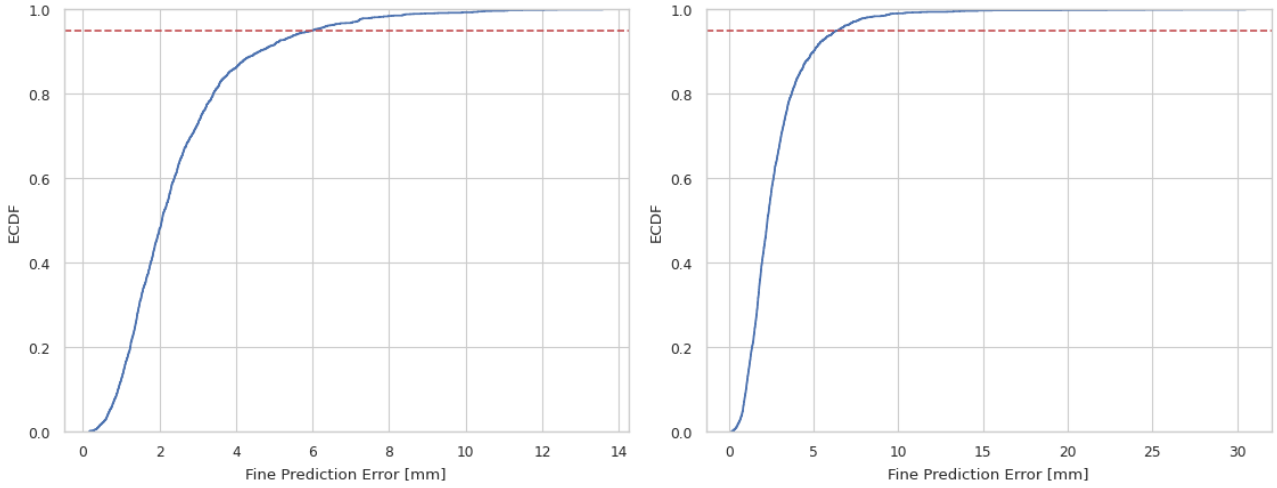



**Figure 5.1** Comparative CDFs for validation and testing errors, with the 95th percentile marked.

| Group | N POIs in Val | N POIs in Test | Max Error Val | Max Error Test |
|---|---|---|---|---|
| Outer | 74 | 64 | 13.60 | 23.38 |
| ITL | 19 | 35 | 12.4 | 8.84 |
| Spinous | 5 | 4 | 6.27 | 7.47 |
| Inner | 1 | 8 | 7.24 | 25.61 |

**Table 5.8** Number of POIs per group in the top-5% errors for the validation and test set and the maximum error per group.

We visually inspect these outliers to understand the underlying reasons. On the validation set, we inspect the outliers in the Inner and Spinous groups. In both, our model showed clear deviations from the expected landmark localization. In the ITL group, we identified 6 vertebrae in the same subject where both the right and left ITL were incorrectly annotated on the articular process instead of the transverse process (see Figure 5.2). We marked these POIs as missing in addition to the POIs that failed the quality analysis in data cleaning. The analysis of the Outer group showed less conclusive results. This is mainly because the outer edges of the ligaments, specifically in the case of the ALL, lack clearly defining landmarks on the vertebra, indicating their extension in the lateral dimension. This leads to a certain level of ambiguity in identifying these POIs. Figure 5.3 compares the prediction of the right cranial ALL attachment point as predicted by our model and annotated in the dataset. We included a reference image with the ground truth of the same POI type on the same vertebra but for a different subject. Despite the very large distance to the ground truth, we believe our model's prediction is reasonable. Similarly, the height of the ITL varies between annotations, leading to high errors even if the prediction aligns well with other annotations. In Appendix A, we include several more examples stemming from the top-error group listed in Table 5.8 with prediction, ground truth, and reference ground truth annotations. Our qualitative analysis leads us to believe that our model produces reasonable results despite a high indicated error in many instances.

In the test set, the visual inspection revealed that one subject remained in the test set despite showing a global misalignment between the segmentation and the marked POIs despite our efforts in the data cleaning (see Figure 5.4). This was because it occurred on a much smaller scale, leaving the surface distances and spatial configurations intact in their majority but still introducing significant shifts producing ill-annotated landmarks. We dropped this subject from the test set. Further, three of the four outliers in the Spinous group could be clearly identified as marked incorrectly in the ground truth and included in the list of missing annotations. The annotations of four vertebrae in one subject exhibited the same errors for the ITL as already observed in the validation set, being marked on the articular process instead of the

**Figure 5.2** Prediction (left) and ground truth (middle) of a subject with wrongly annotated ITL attachment points on a C3. The ground truth for the C3 of a different subject is shown on the right for reference.

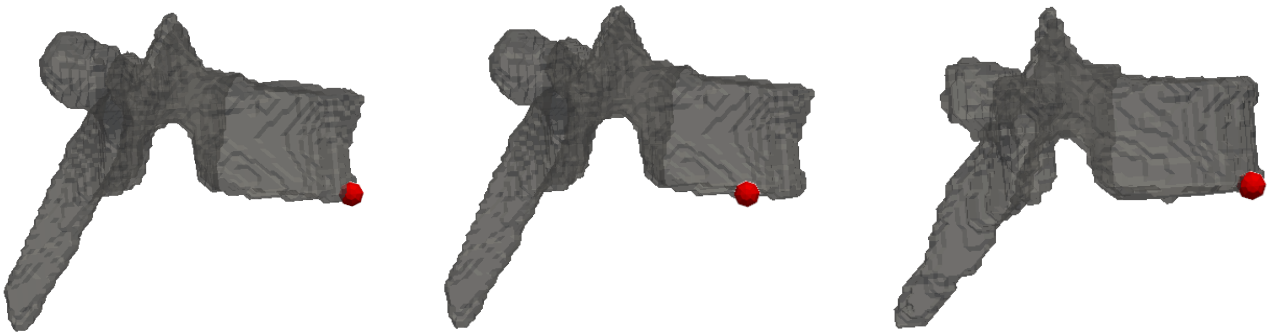


**Figure 5.3** Prediction (left) and ground truth (middle) of the ALL_CR_D on a T6. The ground truth for the same POI on the T6 of a different subject is shown on the right for reference.

transverse process and were marked missing. The POIs in the inner group showed similar ambiguities as those in the validation set.

### Re-Evaluating Registration Baseline and Model

Having identified some erroneous annotations, we re-evaluated the registration baseline and our model to understand the respective capabilities better. In Table 5.9, we compare the mean, median, MSE and accuracy of the surface-projected registration results and fine outputs of our model, representing the best predictions, respectively. The better of the metrics within the same set is marked in bold.

We again compare the model's coarse and fine performance to obtain more reliable insights into the refinement module's contribution in Table 5.10.

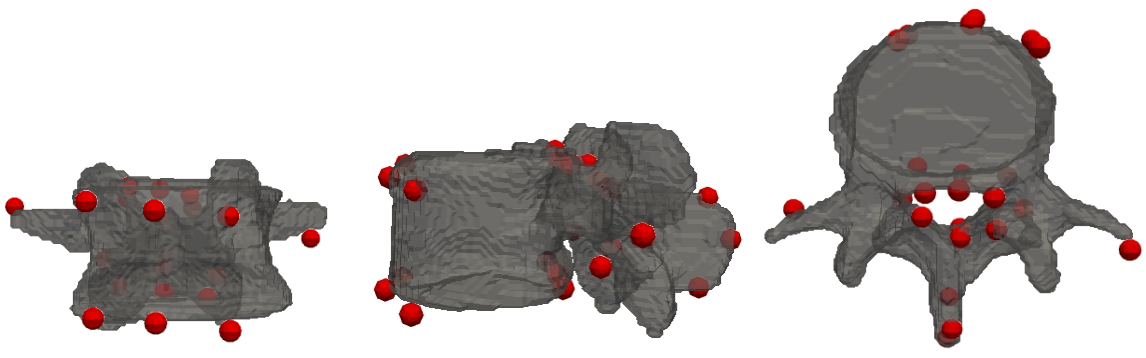


**Figure 5.4** Mis-aligned ground truth on a subject originally included in the test set as viewed from front, side, and above.

| Split | Prediction Type | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | Registration | 3.04 | 2.45 | **4.15** | 0.38 |
| | Neural Network (NN) Model | **2.17** | **1.73** | 7.52 | **0.51** |
| Test | Registration | 2.96 | 2.45 | **3.61** | 0.36 |
| | NN Model | **2.19** | **2.00** | 7.36 | **0.50** |

**Table 5.9** Comparison of registration-based and learned POIs after removing wrong annotations.

| Split | Prediction Type | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | Coarse | 2.38 | 2.24 | 8.09 | 0.42 |
| | Fine | **2.17** | **1.73** | **7.52** | **0.51** |
| Test | Coarse | 2.32 | 2.24 | 7.75 | 0.43 |
| | Fine | **2.19** | **2.00** | **7.36** | **0.50** |

**Table 5.10** Comparison of coarse and fine predictions after removing wrong annotations.

### 5.1.4 Training on Ligament Attachment Points in the Sagittal Plane

Our analysis of the worst cases of the model, in particular, the visual inspection, leads us to believe that the outer attachment points of ALL, PLL, and FL, as well as both of the ITL attachment points are not as clearly defined as the remaining ones, which lie in the sagittal plane. To investigate whether these POIs hinder training by providing unreliable gradient updates, we train a model on the sagittal POIs only and compare its performance to our previous model when only considering the sagittal POIs (see Table 5.11.

| Split | Trained On | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | Sagittal | **1.57** | 1.41 | **3.61** | **0.67** |
| | Full | 1.68 | 1.41 | 4.21 | 0.63 |
| Test | Sagittal | **1.59** | 1.41 | **3.77** | **0.66** |
| | Full | 1.77 | 1.41 | 4.55 | 0.59 |

**Table 5.11** Performance Metrics after Training on the LiAP Dataset, restricted to POIs in the Sagittal Plane.

### 5.1.5 Self-Training

Our previous experiments have shown significant noise in the ground truth annotations, which we expect to also occur in the training set. Further, we removed many annotations in our data cleaning steps, potentially limiting the training performance. To incorporate as much data as possible while reducing the noise, we experiment with the following self-training [3] strategy:

We train the model using 5-fold cross-validation to obtain predictions for all samples. After completion, we create new pseudo-annotations by taking the model predictions on the respective validation sets in instances where we had previously marked the POIs as missing. For the remaining POIs, which can still show a significant amount of unclean data as seen in the worst-case analysis, we remove instances where the ground truth and model prediction differ by more than a threshold of 3mm. Similar techniques of filtering the training data based on a disagreement between the model and ground truth for noisy data have been explored before by Nguyen *et al.* [66]. After completing these steps, we re-train a model on the new filtered data enriched with the pseudo-annotations. We summarize our results in Table 5.12.

| Split | Prediction Type | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) |
|---|---|---|---|---|---|
| Val | Coarse | 2.08 | 2.24 | 5.38 | 0.44 |
| | Fine | **1.66** | **1.41** | **3.72** | **0.62** |
| Test | Coarse | 2.84 | 2.45 | 13.00 | 0.35 |
| | Fine | **2.63** | **2.24** | **12.02** | **0.44** |

**Table 5.12** Metrics on the cleaned validation and test set with self-training. Note that the validation set underwent cleaning steps based on the agreement of an initial model with the ground truth, while the test set did not.

## 5.2 Experiments on the Pedicle Screw Dataset

We train our model again on the PS Dataset to see how well the model can adapt to this different task. Since the direction of the pedicle screw is of great importance for accurate placement, we report the mean angular deviation between the trajectory as obtained from the predicted POIs in addition to the distance-based metrics in Table 5.13.

| Split | Pred.Type | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) | Mean Ang. Deviation (°) |
|---|---|---|---|---|---|---|
| Val | Coarse | 7.65 | 7.49 | 62.68 | 0.00 | |
| | Fine | **3.90** | **3.38** | **19.48** | **0.20** | 9.14 |
| Test | Coarse | 7.25 | 6.51 | 59.53 | 0.00 | |
| | Fine | **5.43** | **4.84** | **35.06** | 0.00 | 11.94 |

**Table 5.13** Performance Metrics after training on the PS Dataset.

## 5.3 Training Jointly on Ligament Attachment Point and Pedicle Screw Datasets

The results on the PS Dataset show poor performance compared to the LiAP Dataset data. Given the small size of the dataset, we aim to incorporate additional data from the LiAP Dataset. Hence, we believe the model may benefit from a multi-task learning approach [97]: We train the model to predict the ligament landmarks and the screw trajectory key points simultaneously. Since we have two disjunct datasets, we adopt the following strategy: All POI types are predicted for a given input. The ground truth for the POIs that are not present is marked missing in the same way as for the annotations removed during data cleaning, such that the loss is masked to only backpropagate on the existing annotations.

- **Feature Extraction** Predicting the ligament attachment points in addition to the screw trajectory points requires a high level of semantic understanding, potentially encouraging the model to learn more descriptive features.
- **Attention Context** Our refinement module relates landmarks to each other. Accurate predictions of the ligament points may provide a reference frame to rationalize the screw trajectory points better.

Due to the difference in the number of samples, we adopt a weighted random sampling scheme, ensuring that the datasets are balanced during the training process. We report the results of this experiment in Table 5.14.

| Split | Pred. Type | Mean Dist. | Median Dist. | MSE | Accuracy (2mm) | Mean Ang. Deviation (°) |
|---|---|---|---|---|---|---|
| Val | Coarse | 6.62 | 5.98 | 53.30 | 0.00 | |
| | Fine | **5.96** | **5.36** | **43.89** | 0.00 | 14.27 |
| Test | Coarse | **6.41** | **6.32** | **47.53** | 0.05 | |
| | Fine | 6.65 | 6.96 | 50.32 | 0.05 | 20.48 |

**Table 5.14** Performance on the PS Dataset after joint training, incorporating the LiAP Dataset.

# 6 Discussion

In this chapter, we interpret and discuss the results of our experiments. We begin by reviewing our experiments, comparing them to existing literature, and contextualizing the results. Next, we will discuss our model's strengths and limitations. Finally, we will discuss the tasks' data representation and framing.

## 6.1 Interpretation Of Results

This section is dedicated to the interpretation of our results. Following the order of the conducted experiments, we will review the results obtained on the LiAP Dataset, followed by the PS Dataset, including the approach of multi-task Training with the added LiAP Dataset.

**Comparison to Literature** Quantitative performances in anatomical landmark detection are notoriously different due to the vast array of considered anatomical structures, landmarks, and image modalities [89]. Li *et al.* report a mean distance of 1.86mm and 2.07mm between automatically derived landmarks on vertebrae via a hand-crafted algorithm operating on a surface mesh [56]. The landmarks considered in their work do not represent ligament attachment points but geometrical key points. A closely related and more intensely studied task to ours is cephalometric landmark detection, involving the identification of POIs on the skull and jaw. Recently, Chen *et al.* reported an average detection error of 1.64mm and 2.37mm, respectively, on two different datasets of 3D Cone-Beam CT images in this task. Kang *et al.* achieved a mean error of 1.96mm on 3D CT scans. Our results are comparable on the LiAP Dataset based on mean error alone. However, it is notable that cephalometry, in general, deals with much larger inputs, e.g., the scans used by Chen *et al.* had a resolution of $768 \times 768 \times 576$ compared to the $128 \times 128 \times 96$ used in this work. This introduces particular challenges regarding computational feasibility and typically requires significant downscaling of the input. While our method is adaptable to such a setting, as the coarse module can consume downscaled inputs while maintaining full resolution in the refinement stage, it remains to be seen whether comparable performance can be attained.

In a recent study concerning automated pedicle screw placement, Scherer *et al.* compared a traditional atlas-based method with a NNs method [79]. Using a U-Net-based architecture, they treated the problem as a segmentation problem, predicting on a voxel-wise level whether a coordinate belongs to the trajectory directly on the spine CT. They reported a minimum average distance over two manual annotations of 3.93mm and 3.49 mm of the screw head and tip points, respectively, with a mean angular deviation of 4.46° and a standard deviation of 2.86°. The atlas-based method achieved 7.77mm and 7.81 mm for the head and tip with an angular deviation of 6.70° and 3.53° standard deviation. Notably, their training dataset was much larger than ours, comprising 155 spine CT images compared to the six used in our work.

**Comparison To Registration Baseline** Our initial model shows an improvement over the baseline in terms of mean error, median error, and accuracy. Interestingly, the MSE of the registration-based predictions was lower than the one of the model. It is notable in this context that in the final comparison of our model and the baseline, we removed outliers based on the surface distance from the baseline, which we did not do for our model. However, even before outlier removal, the registration MSE is lower than for the learned POI locations, indicating a lower tendency to produce far-off results. This may be due to an inherent strength of registration-based prediction: Because the locations are transformed locations of well-structured POI locations, the spatial consistency is inherently respected. Essentially, the configuration of POI in the reference images encodes prior high-level anatomical information. Thus, while the adaptability of registration to finer anatomical structures is low, limiting the ability to generate finely tuned results,

registration tends to provide results in a reasonable range of the ground truth. The NN model does not have this advantage, resulting in a higher chance to produce errors on a grander scale even though most predictions are more accurate, as indicated by the improved mean and median errors and accuracy at a 2mm threshold.

**Training on POIs in the Sagittal Plane** Training on a subset of the POIs is a significant simplification of the task. It can, therefore, be expected to improve the model's performance even when all annotations are well-defined and accurate. However, paired with our qualitative analysis and the significant improvement in generalization ability to the test set, the results of the experiments support our suspicion that the lateral POIs of the ALL, PLL and FL lack a clearly defining characteristic and introduce noise to the training process.

**Self-Training** Similarly to the case with Training on POIs in the sagittal plane, the interpretation of the self-training results is not straightforward. The improvement of the metrics on the validation set is substantial. However, we need to consider that the data-cleaning steps performed based on the initial model's predictions cannot distinguish between a high error due to high noise in the annotations or a genuinely erroneous output of the model. This technique, therefore, suppresses samples that the model struggles to predict even though their annotations are correct. However, while we can not assume the model will produce objectively better results based on the evaluation, the increase in performance hints at the ability of the model to learn the underlying pattern of the cleaned predictions even though they include a bias.

**Multi-Task Training** Due to the typical lack of large-scale fully annotated datasets, increased attention has been received by methods incorporating other datasets or labels that may be available [12]. Our approach to training on the two datasets jointly while masking the loss to backpropagate only on available annotations fits well into this overall trend. However, this technique could not improve the model's performance on the PS Dataset. Notably, due to the minimal dataset, only a single subject is used for validation and testing, significantly limiting the robustness of the gathered metrics in performance evaluation. However, we also believe that a contributing factor to the lack of performance improvement is a difference in underlying data distributions: As the PS Dataset reflects a real-world clinical scenario, the number of pathological cases with accompanying altered morphologies in the vertebrae was significantly higher when compared to the LiAP Dataset, limiting the ability to benefit from learning a common feature representation.

**Coarse and Fine Predictions** We designed the refinement module of our model specifically intending to address the issue above, i.e., model spatial relationships between landmarks and correct them accordingly. The comparison to the registration baseline shows significant potential for improvement in this regard, as a high MSE indicates still a high number of outliers in the prediction. However, we note that the refined predictions consistently outperform the coarse predictions across all experiments, indicating the potential of our approach. The tendency was particularly pronounced in the validation data of the self-training approach, indicating a good capability of the refinement module to model the more regularly structured cleaned labels and pseudo-labels.

## 6.2 Limitations and Future Directions

**Model Design** Our model uses a standard CNN architecture to extract a feature map from the semantic segmentation mask. Sekuboyina *et al.* have successfully used a point cloud representation of only the vertebra surface derived from segmentation masks in encoding vertebra shapes for the downstream task of fracture detection [81]. Conversely, Graham *et al.* [29] developed Submanifold Sparse Convolution (SSC) in 3D scene understanding, a convolutional operator modified to process spatially sparsely populated regular grids as an alternative to point cloud processing methods. The SSC operator maintains spatial

sparsity by applying the convolutional kernel only if its center is non-zero in the input to explicitly model learning on data that lies inherently on submanifolds, like curved lines in 2D images or curved surfaces in 3D space. This saves substantial computational costs, as zero elements carry no computational overhead in the implementation while achieving competitive performance. We believe the integration of SSC carries the potential to adapt our method to other tasks like cephalometric landmark detection, where the larger input sizes require explicit consideration of computational costs.

Further, our model uses an embedding of the vertebra type, motivated by the fact that the spatial configuration of the POIs depends on the morphology of the vertebra, which in turn differs between different vertebra types. However, even vertebrae of the same kind can exhibit significant morphological differences between subjects. A more nuanced approach could be to integrate a learned representation of the morphology of the individual vertebra extracted in a separate branch from the landmark-wise feature extraction, possibly leveraging transfer learning by employing an encoder of a model pre-trained for reconstruction. Whether such refinements of representing the vertebra can improve the performance of our model remains to be seen.

**Data and Framing of the Tasks** Many of our findings point towards the data quality, representing a limiting factor in our work. The data cleaning revealed a significant amount of implausible data, and our later experiments and evaluations showed that not all of it was detected with these steps. Further, we see a considerable improvement of the metrics when restricting the task to the subset that we believe to have the most robust definitions, the POIs in the sagittal plane, and when adopting a self-training approach cleaning the data based on agreement between an initial model and ground truth. We believe that the comparably poor performance on POIs representing the edges of ALL, PLL, and FL, as well as the POIs representing the ITL, is in large parts caused by a lack of clear definition of where to place these points in the CT scan, leading us to suspect that there may be high intra-operator variance in annotations. However, we do not have data available to support this possibility. Developing specialized methods for the overarching goal of biomedical simulations may need to incorporate demand-specific prior knowledge to handle the deviations or aim to collect more consistently annotated data.

In the case of automation of pedicle screw placement, we believe a purely landmark-based approach to be limited. Firstly, there is a lack of a precise, generally accepted definition of crucial landmarks; secondly, our strategy was based on predicting points along the trajectory. However, this reveals an inadequate representation of the requirements for proper screw placement: while shifts along the trajectory make no difference, even a tiny medial shift can lead to the breaching of the screw into the spinal canal and thus pose a significant problem. An optimization approach that aims to minimize the distance to arbitrarily defined points cannot adequately represent this. We consider it more goal-oriented to identify an anchor point in the pedicle and predict a direction vector. However, current annotations do not include such anchor points. They are difficult to obtain automatically from the head and tips due to the trajectory length within the pedicle differing between subjects and vertebra types. Furthermore, although our model considers the configuration of points to each other, we do not explicitly supervise it, for instance, with a loss term that promotes coplanarity of the points or specific angles of the screws relative to each other or the endplate.

For both tasks, we believe synthetic data could enhance the model's ability and provide a way to address the limited amount of available data while reducing bias in the annotations [71].

# 7 Conclusion

We developed a novel neural architecture for predicting Points of Interest (POIs) on anatomical structures, capitalizing on recent advancements in spatial attention and transformer architectures in computer vision. Instead of using direct CT input, we opted for semantic segmentation masks, building on prior research highlighting the advantages of learning from morphological representations over traditional grayscale images. Our work has set a baseline for detecting ligament attachment points on human vertebrae, achieving accuracy comparable to state-of-the-art methods in cephalometric landmark detection. To our knowledge, it marks the first model of its kind for this task.

Our research also explores the potential for using this model to expedite surgical screw placement planning. However, we encountered significant limitations due to the limited amount of data. Our attempts to counteract this issue with a multi-task learning approach to leverage the much bigger dataset for ligament attachment point predictions showed no improvement over the single-task learning approach on the much smaller dataset. We believe the differing data distributions negatively affect the applicability of multi-task learning in our case, with the pedicle screw placement dataset containing a significant amount of pathological cases as it stems from a real-world clinical scenario.

In addressing the prediction of ligament attachment points, the lack of objective criteria for marking POIs on the outer edges of ligaments and noisy data posed additional challenges. To explore these issues, we conducted two experiments: training on a subset of points that shows the least variability and implementing a self-learning approach to filter outliers from the annotations. These experiments suggest that our model can more effectively learn patterns from cleaned data than from original annotations. However, the absence of manually cleaned data precludes confirmation that these patterns adhere to clinical standards.

Looking forward, we see multiple opportunities to enhance our method. Inspired by Sekuboyina *et al.* [81], who used a point cloud derived from segmentation masks to encode vertebra shapes, we believe adopting a similar approach could refine our feature extraction process. Alternatively, the Sparse Submanifold Convolutions (SSC) developed by Graham *et al.* [29] offer a promising alternative for processing spatially sparse grids, which could be superior to current point cloud processing methods. To overcome the current limitations in data quality and annotation consistency, synthetic data could be strategically employed to enhance the availability and consistency of training datasets.

Moreover, our research underscores the need to expand annotations for pedicle screw placement to include clinically critical landmarks, such as a screw trajectory keypoint within the narrow bone corridor in the pedicle, vital for successful surgical outcomes. By enhancing these annotations and incorporating supervision techniques related to quality assessment criteria directly, future research could substantially improve the accuracy and clinical applicability.

In conclusion, our groundwork demonstrates significant potential for surgical planning and improving subject-specific spine simulations. These can help expand the currently limited understanding of contributing factors to lower back pain. At the same time, our research highlights the need for objective and requirement-specific annotations to exhaust their full potential.

# A

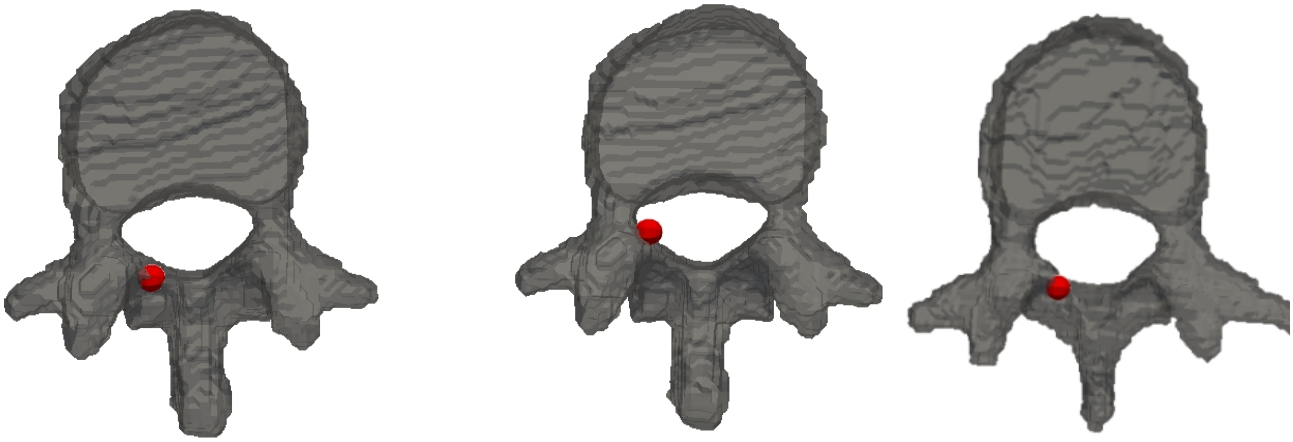


**Figure A.1** Comparison of model prediction (left) and ground truth (middle) of an FL_CR_S on a L1. The ground truth for the same POI on the same vertebra type of a different subject is shown on the right for reference.

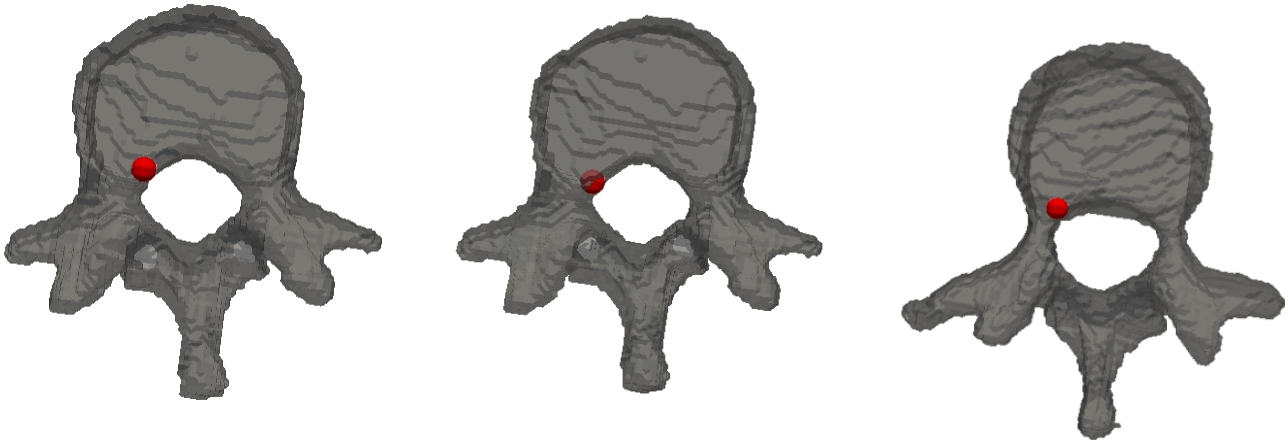


**Figure A.2** Comparison of model prediction (left) and ground truth (middle) of an FL_CR_D on a L1. The ground truth for the same POI on the same vertebra type of a different subject is shown on the right for reference.

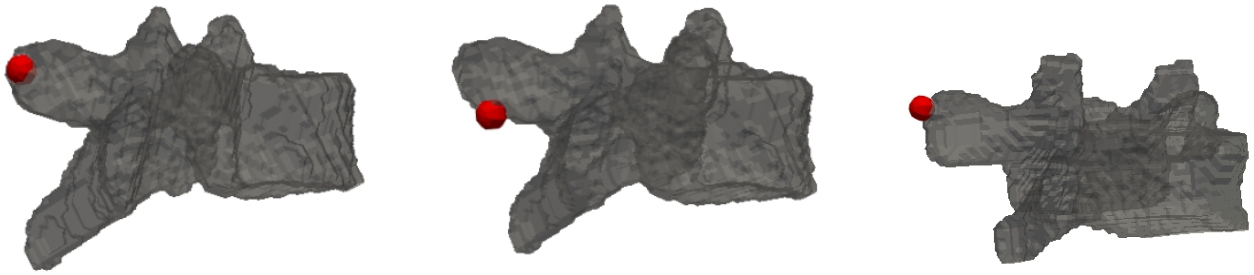


**Figure A.3** Comparison of model prediction (left) and ground truth (middle) of an ITL_D on a T3. The ground truth for the same POI on the same vertebra type of a different subject is shown on the right for reference.

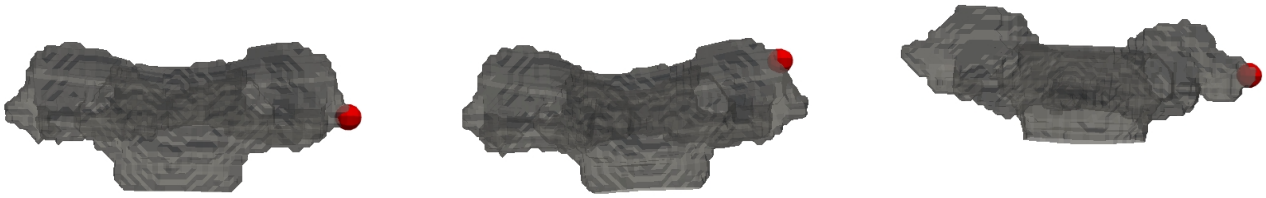


**Figure A.4** Comparison of model prediction (left) and ground truth (middle) of an ITL_S on a T3. The ground truth for the same POI on the same vertebra type of a different subject is shown on the right for reference.

**Figure A.5** Comparison of model prediction (left) and ground truth (middle) of an ALL_CR_S on a T8. The ground truth for the same POI on the same vertebra type of a different subject is shown on the right for reference.

# List of Tables

# List of Figures

# Bibliography


[1] Md. Zahangir Alom, C. Yakopcic, Mahmudul Hasan, et al. “Recurrent residual U-Net for medical image segmentation”. In: *Journal of Medical Imaging* 6 (2019), pp. 014006–014006. DOI: 10.1117/1.JMI.6.1.014006.

[2] Neena Aloysius and M. Geetha. “A review on deep convolutional neural networks”. In: *2017 International Conference on Communication and Signal Processing (ICCSP)* (2017), pp. 0588–0592. DOI: 10.1109/ICCSP.2017.8286426.

[3] Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, et al. “Self-training: A survey”. In: *arXiv preprint arXiv:2202.12040* (2022).

[4] Melisa Ankut, Arda Mamur, and Daniel-Jordi Regenbrecht. *3D Castellvi Prediction*. July 2023. URL: https://github.com/ardamamur/3D-Castellvi-Prediction.

[5] Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, et al. “Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction”. In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 37–52. DOI: 10.1109/MSP.2017.2762355.

[6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2014.

[8] Dulari Bhatt, Chirag Patel, Hardik Talsania, et al. “CNN variants for computer vision: History, architecture, application, challenges and future scope”. In: *Electronics* 10.20 (2021), p. 2470.

[9] Gianni Brauwers and F. Frasincar. “A General Survey on Attention Mechanisms in Deep Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 35 (2022), pp. 3279–3298. DOI: 10.1109/TKDE.2021.3126456.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, et al. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[11] Runnan Chen, Yuexin Ma, Nenglun Chen, et al. “Structure-Aware Long Short-Term Memory Network for 3D Cephalometric Landmark Detection”. In: *IEEE Transactions on Medical Imaging* 41 (2021), pp. 1791–1801. DOI: 10.1109/TMI.2022.3149281.

[12] Veronika Cheplygina, Marleen de Bruijne, and Josien P.W. Pluim. “Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis”. In: *Medical Image Analysis* 54 (2019), pp. 280–296. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2019.03.009. URL: https://www.sciencedirect.com/science/article/pii/S1361841518307588.

[13] Marina Codari, Matteo Caffini, Gianluca M. Tartaglia, et al. “Computer-aided cephalometric landmark annotation for CBCT Data”. In: *International Journal of Computer Assisted Radiology and Surgery* 12.1 (June 2016), pp. 113–121. DOI: 10.1007/s11548-016-1453-9.

[14] Wikimedia Commons. *File:Vertebra Superior View-en.svg — Wikimedia Commons, the free media repository*. [Online; accessed 25-April-2024]. 2023. URL: %5Curl%7Bhttps://commons.wikimedia.org/w/index.php?title=File:Vertebra_Superior_View-en.svg&oldid=818351263%7D.

[15] T. Dao, P. Pouletaut, F. Charleux, et al. “Multimodal medical imaging (CT and dynamic MRI) data and computer-graphics multi-physical model for the estimation of patient specific lumbar spine muscle forces”. In: *Data Knowl. Eng.* 96 (2015), pp. 3–18. DOI: 10.1016/j.datak.2015.04.001.

[16] Yuzhen Ding, Hongying Feng, Yunze Yang, et al. “Deep-learning based fast and accurate 3D CT deformable image registration in lung cancer”. In: *Medical physics* 50.11 (2023), pp. 6864–6880.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *International Conference on Learning Representations* (2021).

[18] A. Esteva, Katherine Chou, Serena Yeung, et al. “Deep learning-enabled medical computer vision”. In: *NPJ Digital Medicine* 4 (2021). DOI: 10.1038/s41746-020-00376-2.

[19] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, et al. “Wing loss for robust facial landmark localisation with convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2235–2245.

[20] Jie Fu, K. Singhrao, X. Qi, et al. “3D multi-path DenseNet for improving automatic segmentation of glioblastoma on pre-operative multi-modal MR images.” In: *Medical physics* (2021). DOI: 10.1002/mp.14800.

[21] R. Gaines. “The Use of Pedicle-Screw Internal Fixation for the Operative Treatment of Spinal Disorders*”. In: *The Journal of Bone I& Joint Surgery* 82 (2000), p. 1458. DOI: 10.2106/00004623-200010000-00013.

[22] Fabio Galbusera and Hans-Joachim Wilke. *Biomechanics of the spine: Basic concepts, spinal disorders and treatments*. en. San Diego, CA: Academic Press, Apr. 2018.

[23] Pengcheng Gao, K. Lu, Jian Xue, et al. “A Coarse-to-Fine Facial Landmark Detection Method Based on Self-attention Mechanism”. In: *IEEE Transactions on Multimedia* 23 (2021), pp. 926–938. DOI: 10.1109/TMM.2020.2991507.

[24] Tobias Gass, Gábor Székely, and Orcun Goksel. “Multi-atlas Segmentation and Landmark Localization in Images with Large Field of View”. In: *MCV*. 2014. URL: https://api.semanticscholar.org/CorpusID:154774.

[25] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Nov. 2011, pp. 315–323. URL: https://proceedings.mlr.press/v15/glorot11a.html.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[27] Krzysztof J Gorgolewski, Tibor Auer, Vince D Calhoun, et al. “The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments”. In: *Scientific data* 3.1 (2016), pp. 1–9.

[28] Ardeshir Goshtasby. *Image registration principles, tools and methods*. Springer London, 2012.

[29] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. “3d semantic segmentation with submanifold sparse convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9224–9232.

[30] Changlu Guo, Marton Szemenyei, Yugen Yi, et al. “SA-UNet: Spatial Attention U-Net for Retinal Vessel Segmentation”. In: *2020 25th International Conference on Pattern Recognition (ICPR)* (2020), pp. 1236–1242. DOI: 10.1109/ICPR48806.2021.9413346.

[31] Benjamín Gutiérrez Becker. “Machine Learning Methods for Computer Assisted Diagnosis and Medical Image Registration”. en. PhD thesis. Technische Universität München, 2019, p. 96. URL: https://mediatum.ub.tum.de/1473771.

[32] Kun Han, Shanlin Sun, Xiangyi Yan, et al. “Diffeomorphic image registration with neural velocity field”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 1869–1879.

[33] P. Hanson and S. Magnusson. “The difference in anatomy of the lumbar anterior longitudinal ligament in young African-Americans and Scandinavians.” In: *Archives of physical medicine and rehabilitation* 79 12 (1998), pp. 1545–8. DOI: 10.1016/S0003-9993(98)90417-8.

[34] Robert M. Haralick, Stanley R. Sternberg, and Xinhua Zhuang. “Image Analysis Using Mathematical Morphology”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.4 (1987), pp. 532–550. DOI: 10.1109/TPAMI.1987.4767941.

[35] J. Hartvigsen, M. Hancock, A. Kongsted, et al. “What low back pain is and why we need to pay attention”. In: *The Lancet* 391 (2018), pp. 2356–2367. DOI: 10.1016/S0140-6736(18)30480-X.

[36] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, et al. “Unetr: Transformers for 3d medical image segmentation”. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2022, pp. 574–584.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[38] D. Hoy, L. March, P. Brooks, et al. “The global burden of low back pain: estimates from the Global Burden of Disease 2010 study”. In: *Annals of the Rheumatic Diseases* 73 (2014), pp. 968–974. DOI: 10.1136/annrheumdis-2013-204428.

[39] Gao Huang, Zhuang Liu, Laurens van der Maaten, et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[40] N. Ibtehaz and Mohammad Sohel Rahman. “MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation”. In: *Neural networks : the official journal of the International Neural Network Society* 121 (2019), pp. 74–87. DOI: 10.1016/j.neunet.2019.08.025.

[41] Hiroko Ikeuchi and Ko Ikuta. “Accuracy of pedicle screw insertion in the thoracic and lumbar spine: a comparative study between percutaneous screw insertion and conventional open technique”. In: *Archives of Orthopaedic and Trauma Surgery* 136 (2016), pp. 1195–1202.

[42] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html.

[43] F. Jiang, Yong Jiang, Hui Zhi, et al. “Artificial intelligence in healthcare: past, present and future”. In: *Stroke and Vascular Neurology* 2 (2017), pp. 230–243. DOI: 10.1136/svn-2017-000101.

[44] R. Johnson, E. Crelin, A. White, et al. “Some new observations on the functional anatomy of the lower cervical spine.” In: *Clinical orthopaedics and related research* 111 (1975), pp. 192–200. DOI: 10.1097/00003086-197509000-00027.

[45] Uday Kamath, Kenneth L. Graham, and Wael Emara. *Transformers for machine learning: A deep dive*. CRC Press, Taylor & Francis Group, 2022.

[46] Sung Ho Kang, Kiwan Jeon, Sang-Hoon Kang, et al. “3D cephalometric landmark detection by multiple stage deep reinforcement learning”. In: *Scientific Reports* 11.1 (Sept. 2021). DOI: 10.1038/s41598-021-97116-7.

[47] Salman Hameed Khan, Muzammal Naseer, Munawar Hayat, et al. “Transformers in Vision: A Survey”. In: *ACM Computing Surveys (CSUR)* 54 (2021), pp. 1–41. DOI: 10.1145/3505244.

[48] Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, et al. “Pure Transformers are Powerful Graph Learners”. In: *ArXiv* abs/2207.02505 (2022). DOI: 10.48550/arXiv.2207.02505.

[49] M. Kirby, T. A. Sikoryn, D. Hukins, et al. “Structure and mechanical properties of the longitudinal ligaments and ligamentum flavum of the spine.” In: *Journal of biomedical engineering* 11 3 (1989), pp. 192–6. DOI: 10.1016/0141-5425(89)90139-8.

[50] V. Kosmopoulos and C. Schizas. “Pedicle Screw Placement Accuracy: A Meta-analysis”. In: *Spine* 32 (2007), E111–E120. DOI: 10.1097/01.brs.0000254048.79024.8b.

[51] A. Krizhevsky, I. Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60 (2012), pp. 84–90. DOI: 10.1145/3065386.

[52] Ramesh Kumar, Jaims Lim, R. Mekary, et al. “Traumatic Spinal Injury: Global Epidemiology and Worldwide Volume.” In: *World neurosurgery* 113 (2018), e345–e363. DOI: 10.1016/j.wneu.2018.02.033.

[53] Y. LeCun, B. Boser, J. S. Denker, et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[54] Yann LeCun, Léon Bottou, Yoshua Bengio, et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[55] Tanja Lerchl, Kati Nispel, T. Baum, et al. “Multibody Models of the Thoracolumbar Spine: A Review on Applications, Limitations, and Challenges”. In: *Bioengineering* 10 (2023). DOI: 10.3390/bioengineering10020202.

[56] Bo Li, Wentao Yu, Junhua Zhang, et al. “An automatic method for landmark identification of the 3D vertebrae”. In: *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*. ICCAI ’22. Tianjin, China: Association for Computing Machinery, 2022, pp. 400–406. ISBN: 9781450396110. DOI: 10.1145/3532213.3532273. URL: https://doi.org/10.1145/3532213.3532273.

[57] Hui Li, Zidong Guo, Seon-Min Rhee, et al. “Towards Accurate Facial Landmark Detection via Cascaded Transformers”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 4166–4175. DOI: 10.1109/CVPR52688.2022.00414.

[58] Zewen Li, Fan Liu, Wenjie Yang, et al. “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33 (2020), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.

[59] I. Lieberman, S. Kisinde, and Shea Hesselbacher. “Robotic-Assisted Pedicle Screw Placement During Spine Surgery.” In: *JBJS essential surgical techniques* 10 2 (2020), e0020. DOI: 10.2106/jbjs.st.19.00020.

[60] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. 2017. URL: https://api.semanticscholar.org/CorpusID:53592270.

[61] Diogo C. Luvizon, Hedi Tabia, and David Picard. “Human pose regression by combining indirect part detection and contextual information”. In: *Computers & Graphics* 85 (2019), pp. 15–22. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2019.09.002. URL: https://www.sciencedirect.com/science/article/pii/S0097849319301475.

[62] Di Meng, Edmond Boyer, and Sergi Pujades. “Vertebrae localization, segmentation and identification using a graph optimization and an anatomic consistency cycle”. In: *Computerized Medical Imaging and Graphics* 107 (2023), p. 102235. ISSN: 0895-6111. DOI: https://doi.org/10.1016/j.compmedimag.2023.102235. URL: https://www.sciencedirect.com/science/article/pii/S0895611123000538.

[63] P. Merloz, J. Tonetti, L. Pittet, et al. “Pedicle Screw Placement Using Image Guided Techniques”. In: *Clinical Orthopaedics and Related Research* NA; (1998), pp. 39–48. DOI: 10.1097/00003086-199809000-00006.

[64] N. Mitsuhashi, K. Fujieda, T. Tamura, et al. “BodyParts3D: 3D structure database for anatomical concepts”. In: *Nucleic Acids Research* 37.Database (Jan. 2009), pp. D782–D785. ISSN: 1362-4962. DOI: 10.1093/nar/gkn613. URL: http://dx.doi.org/10.1093/nar/gkn613.

[65] Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked hourglass networks for human pose estimation”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer. 2016, pp. 483–499.

[66] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, et al. “SELF: Learning to Filter Noisy Labels with Self-Ensembling”. In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HkgsPhNYPS.

[67] L. Nguyen, Van Dung Pham, Yanfen Li, et al. “Facial Landmark Detection With Learnable Connectivity Graph Convolutional Network”. In: *IEEE Access* 10 (2022), pp. 94354–94362. DOI: 10.1109/ACCESS.2022.3200037.

[68] Zhaoyang Niu, G. Zhong, and Hui Yu. “A review on the attention mechanism of deep learning”. In: *Neurocomputing* 452 (2021), pp. 48–62. DOI: 10.1016/J.NEUCOM.2021.03.091.

[69] Julia M. H. Noothout, Bob D. De Vos, Jelmer M. Wolterink, et al. “Deep Learning-Based Regression and Classification for Automatic Landmark Localization in Medical Images”. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 4011–4022. DOI: 10.1109/TMI.2020.3009002.

[70] M. Panjabi, T. Oxland, and Edward H. Parks. “Quantitative anatomy of cervical spine ligaments. Part II. Middle and lower cervical spine.” In: *Journal of spinal disorders* 4 3 (1991), pp. 277–85. DOI: 10.1097/00002517-199109000-00004.

[71] Anthony Paproki, Olivier Salvado, and Clinton Fookes. “Synthetic Data for Deep Learning in Computer Vision & Medical Imaging: A Means to Reduce Data Bias”. In: *ACM Comput. Surv.* (May 2024). Just Accepted. ISSN: 0360-0300. DOI: 10.1145/3663759. URL: https://doi.org/10.1145/3663759.

[72] Christian Payer, Darko Štern, Horst Bischof, et al. “Regressing Heatmaps for Multiple Landmark Localization Using CNNs”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, et al. Cham: Springer International Publishing, 2016, pp. 230–238. ISBN: 978-3-319-46723-8.

[73] Alexander Perdomo-Pantoja, Wataru Ishida, C. Zygourakis, et al. “Accuracy of Current Techniques for Placement of Pedicle Screws in the Spine: A Comprehensive Systematic Review and Meta-Analysis of 51,161 Screws.” In: *World neurosurgery* (2019). DOI: 10.1016/j.wneu.2019.02.217.

[74] David W Polly Jr, Alexandra K Yaszemski, and Kristen E Jones. “Placement of thoracic pedicle screws”. In: *JBJS Essential Surgical Techniques* 6.1 (2016), e9.

[75] V. Puvanesarajah, J. Liauw, S. Lo, et al. “Techniques and accuracy of thoracolumbar pedicle screw placement.” In: *World journal of orthopedics* 5 2 (2014), pp. 112–23. DOI: 10.5312/wjo.v5.i2.112.

[76] V. Ravindra, S.S. Senglaub, A. Rattani, et al. “Degenerative Lumbar Spine Disease: Estimating Global Incidence and Worldwide Volume”. In: *Global Spine Journal* 8 (2018), pp. 784–794. DOI: 10.1177/2192568218770769.

[77] Janosh Riebesell and Stefan Bringuier. *Collection of standalone TikZ images*. Version 0.1.0. 10.5281/zenodo.7486911 - https://github.com/janosh/tikz. Aug. 9, 2020. DOI: 10.5281/zenodo.7486911. URL: https://github.com/janosh/tikz (visited on 05/04/2023).

[78] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[79] Moritz Scherer, Lisa Kausch, Akbar Bajwa, et al. "Automatic Planning Tools for Lumbar Pedicle Screws: Comparison and Validation of Planning Accuracy for Self-Derived Deep-Learning-Based and Commercial Atlas-Based Approaches". In: *Journal of Clinical Medicine* 12.7 (2023). ISSN: 2077-0383. DOI: 10.3390/jcm12072646. URL: https://www.mdpi.com/2077-0383/12/7/2646.

[80] Vincenza Sciortino, S. Pasta, T. Ingrassia, et al. "On the Finite Element Modeling of the Lumbar Spine: A Schematic Review". In: *Applied Sciences* (2023). DOI: 10.3390/app13020958.

[81] Anjany Kumar Sekuboyina, Markus Rempfler, Alexander Valentinitsch, et al. "Probabilistic Point Cloud Reconstructions for Vertebral Shape Analysis". In: *ArXiv* abs/1907.09254 (2019). URL: https://api.semanticscholar.org/CorpusID:198147492.

[82] Shoaleh Shahidi, Ehsan Bahrampour, Elham Soltanimehr, et al. "The accuracy of a designed software for automated localization of craniofacial landmarks on CBCT images". In: *BMC medical imaging* 14 (2014), pp. 1–8.

[83] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[84] Antonio Strauss. *Spinal Anatomy*. Vol. 1st ed. The English Press, 2012. ISBN: 9788132345381. URL: https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=406975&site=ehost-live.

[85] C. Bane Sullivan and Alexander Kaszynski. "PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)". In: *Journal of Open Source Software* 4.37 (May 2019), p. 1450. DOI: 10.21105/joss.01450. URL: https://doi.org/10.21105/joss.01450.

[86] Jonathan Tompson, Arjun Jain, Yann LeCun, et al. "Joint training of a convolutional network and a graphical model for human pose estimation". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 1799–1807.

[87] Tomoki Uemura, J. Näppi, Toru Hironaka, et al. "Comparative performance of 3D-DenseNet, 3D-ResNet, and 3D-VGG models in polyp detection for CT colonography". In: 11314 (2020), pp. 1131435 - 1131435–6. DOI: 10.1117/12.2549103.

[88] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[89] Ching-Wei Wang, Cheng-Ta Huang, Meng-Che Hsieh, et al. "Evaluation and Comparison of Anatomical Landmark Detection Methods for Cephalometric X-Ray Images: A Grand Challenge". In: *IEEE Transactions on Medical Imaging* 34 (2015), pp. 1890–1900. DOI: 10.1109/TMI.2015.2412951.

[90] Fei Wang, Mengqing Jiang, Chen Qian, et al. "Residual attention network for image classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3156–3164.

[91] Ukrit Watchareeruetai, Benjaphan Sommana, Sanjana Jain, et al. "LOTR: Face Landmark Localization Using Localization Transformer". In: *IEEE Access* 10 (2022), pp. 16530–16543. DOI: 10.1109/ACCESS.2022.3149380.

[92] Sanghyun Woo, Jongchan Park, Joon-Young Lee, et al. “Cbam: Convolutional block attention module”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[93] Dong Yang, Shaoting Zhang, Zhennan Yan, et al. “Automated anatomical landmark detection ondistal femur surface using convolutional neural network”. In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. 2015, pp. 17–21. DOI: 10.1109/ISBI.2015.7163806.

[94] N. Yoganandan, S. Kumaresan, and F. Pintar. “Biomechanics of Cervical Spine Ligaments”. In: *Advances in Bioengineering* (1999). DOI: 10.1115/imece1999-0463.

[95] Kun-Hsing Yu, Andrew Beam, and I. Kohane. “Artificial Intelligence in Healthcare”. In: *Artificial Intelligence and Machine Learning for Business for Non-Engineers* (2019). DOI: 10.1201/9780367821654-8.

[96] Marty Zdichavsky, Michael Blauth, Christian Knop, et al. “Accuracy of pedicle screw placement in thoracic spine fractures: part I: inter-and intraobserver reliability of the scoring system”. In: *European Journal of Trauma* 30 (2004), pp. 234–240.

[97] Yu Zhang and Qiang Yang. “An overview of multi-task learning”. In: *National Science Review* 5.1 (Sept. 2017), pp. 30–43. ISSN: 2095-5138. DOI: 10.1093/nsr/nwx105. eprint: https://academic.oup.com/nsr/article-pdf/5/1/30/31567358/nwx105.pdf. URL: https://doi.org/10.1093/nsr/nwx105.

[98] Ziang Zhang, Chengdong Wu, S. Coleman, et al. “DENSE-INception U-net for medical image segmentation”. In: *Computer methods and programs in biomedicine* 192 (2020), p. 105395. DOI: 10.1016/j.cmpb.2020.105395.

[99] Yefeng Zheng, David Liu, Bogdan Georgescu, et al. “3D Deep Learning for Efficient and Robust Landmark Detection in Volumetric Data”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, et al. Cham: Springer International Publishing, 2015, pp. 565–572. ISBN: 978-3-319-24553-9.

[100] Ziliang Zhong, Muhang Zheng, Huafeng Mai, et al. “Cancer image classification based on DenseNet model”. In: *Journal of Physics: Conference Series* 1651 (2020). DOI: 10.1088/1742-6596/1651/1/012143.

[101] Fei Zhu, Sheng Wang, Dun Li, et al. “Similarity attention-based CNN for robust 3D medical image registration”. In: *Biomedical Signal Processing and Control* 81 (2023), p. 104403.